

*MODELING THE SUSTAINABLE USE OF THE BLUE-FRONTED PARROT
(AMAZONA AESTIVA) IN THE DRY CHACO REGION OF ARGENTINA*

Jorge Rabinovich

Centro de Estudios Parasitológicos y de Vectores (CEPAVE), Universidad Nacional de
La Plata, Calle 2 No. 584, e/43 y 44, 1900 La Plata , Prov. de Buenos Aires, Argentina.
E-mail: jorge@ecopaedia.com.ar

A Report to the

Dirección de Fauna Silvestres
(Wildlife Service)

of the

Secretaría de Ambiente y Desarrollo Sustentable
(Secretary of the Environment and Sustainable Development)

of Argentina

Buenos Aires, July, 2004

MODELING THE SUSTAINABLE USE OF THE BLUE-FRONTED PARROT (AMAZONA AESTIVA) IN THE DRY CHACO REGION OF ARGENTINA

Index

Executive Summary

- 1 Introduction
- 2 Study site and present management program
 - 2.1 The region
 - 2.2 Use of the blue-fronted parrot in Argentina
- 3 Methods
 - 3.1 Parameter estimation
 - 3.1.1 Estimation of the intrinsic rate of natural increase (r)
 - 3.1.1.1 The Lotka equation
 - 3.1.1.2 Cole's formula
 - 3.1.1.3 Use of the replacement rate and the generation time
 - 3.1.1.4 Two ages matrix approach
 - 3.1.1.5 Homeotherm meta-analysis
 - 3.1.1.6 Lande's equation
 - 3.1.2 Estimation of the carrying capacity (K)
 - 3.1.2.1 Direct estimation
 - 3.1.2.2 Estimation by transects
 - 3.1.2.3 Estimation by nearest neighbor and point-to-nearest object method
 - 3.1.2.4 Estimation by adaptive sampling
 - 3.1.2.5 Estimation of fledgling density from the harvest area
 - 3.1.2.6 Delimiting the values of parameters r and K based upon actual offtake densities
 - 3.2 The Logistic Population Growth Model
 - 3.3 The "fixed escapement" rule of population management
- 4 Results
 - 4.1 Parameter estimation
 - 4.1.1 The intrinsic rate of natural increase (r)
 - 4.1.1.1 The Lotka equation
 - 4.1.1.2 Estimation of r using Cole's formula
 - 4.1.1.3 Estimation of r using generation time and replacement rate
 - 4.1.1.4 Two ages matrix approach
 - 4.1.1.5 Homeotherm meta-analysis
 - 4.1.1.6 Lande's equation
 - 4.1.2 Estimation of the carrying capacity (K)
 - 4.1.2.1 Direct estimation
 - 4.1.2.2 Estimation by the transect method
 - 4.1.2.3 Estimation by nearest neighbor and point-to-nearest object method
 - 4.1.2.4 Estimation by adaptive sampling
 - 4.1.2.5 Estimation of fledgling density from the harvest area
 - 4.1.2.6 Delimiting the values of parameters r and K based upon actual offtake densities
 - 4.1.3 Summary of parameters' estimations
 - 4.1.3.1 Summary of the intrinsic rate of natural increase (r)
 - 4.1.3.2 Summary of the carrying capacity (K)
 - 4.2 The Logistic Population Growth Model

- 4.3 The “fixed escapement” rule of population management
 - 4.3.1 Behavior of population density under the “*fixed escapement*” rule of population management
 - 4.3.2 The threshold density under the “*fixed escapement*” rule of population management
 - 4.3.3 The optimal offtake density under the “*fixed escapement*” rule of population management
 - 4.3.4 The optimal population offtake under the “*fixed escapement*” rule of population management
 - 4.3.5 Offtake as a proportion of the carrying capacity (K) density
- 4.4 Summary of management results
- 5 Discussion
 - 5.1 The intrinsic rate of natural increase (r)
 - 5.1.1 Survivorship curves
 - 5.1.2 The maternity function
 - 5.1.3 Offspring sex ratio
 - 5.1.4 Cole’s formula, the replacement rate and generation time, and homeotherm meta-analysis
 - 5.1.5 Comparison of the results from the six different methods
 - 5.2 The carrying capacity
 - 5.2.1 The limiting factors
 - 5.2.2 Indicators of the “distance” to the carrying capacity
 - 5.2.3 Density as an approximation to the carrying capacity
 - 5.3 The logistic model of population growth
 - 5.4 The fixed escapement management rule
 - 5.5 Some management aspects
- 6 Conclusions
- Literature Cited
 - Appendix I. Reach and weight data used for the regression to predict reach as a function of weight from eight species of Psittacidae.
 - Appendix II. Regression of weight on reach based upon 361 bird data.
 - Appendix III. Reach and longevity data to predict longevity from regression from 27 species of different bird families.
 - Appendix IV. Reach of the blue-fronted parrot estimated by regression from weight.
 - Appendix V. Excerpts from electronic mail correspondents with experience in *Amazona*.
 - Appendix VI. Report from Eugeni Fernández Nogales on some survival and reproductive features of *Amazona aestiva* under rearing conditions.
 - Appendix VII. Discussion of some literature findings on progeny sex-ratio in birds.
 - Appendix VIII. FORTRAN program developed to simulate the growth and management of a blue-fronted parrot population.
 - Appendix IX. Delimitation of the sampling area for the application of the nearest neighbor and point-to-nearest object methods of density estimation.
 - Appendix X. C++ program for the calculation of the surface area where the random points could land to estimate fledgling density using the the nearest neighbor and point-to-nearest object methods. It also selects a certain number of random points to make a quadrat sampling.

MODELING THE SUSTAINABLE USE OF THE BLUE-FRONTED PARROT (*AMAZONA AESTIVA*) IN THE DRY CHACO REGION OF ARGENTINA

Jorge Rabinovich

Centro de Estudios Parasitológicos y de Vectores (CEPAVE), Universidad Nacional de La Plata, Calle 2 No. 584, e/43 y 44, 1900 La Plata, Prov. de Buenos Aires, Argentina.

E-mail: rabinovi@netverk.com.ar

Executive Summary

1. The “Blue-fronted Parrot Project” (known as “*Proyecto Elé*”) is being carried out by the Secretary of the Environment and Sustainable Development of Argentina, in partnership with the Provinces of Formosa, Chaco, Salta and Jujuy, and with a relationship with the Provinces of Córdoba, Corrientes, Santa Fe, Santiago del Estero and Tucumán. Its main objective is habitat conservation and the sustainable management of the blue-fronted parrot (*Amazona aestiva*) in Argentina, for its commercial use as a pet bird.
2. From 1983 to 1991 around half a million live blue-fronted parrots were authorized by Argentinean provinces for pet trade purposes. This offtake peaked in 1985, with about 75,000 blue-fronted parrot permits, showing afterwards an irregular decrease. On 1996 an experimental management plan was initiated, providing an experience used later to devise a sustainable management plan, which is in effect since 1998 to present.
3. A management model was developed for the Chaco Ecosystem of Argentina, a region with subtropical warm climate, with a mean annual temperature between 18 and 26°C, high potential evapotranspiration (between 900 mm in the South and 1.500 mm in the North), and an annual maximum precipitation of 1.300 mm. The model was developed for a sector of this region called “*Impenetrable*”, which is under strict management, and that covers an area of approximately 20,000 km².
4. A discrete logistic population dynamics model of population growth was used, with population expressed in terms of fledgling density, and its two parameters were estimated: the intrinsic rate of natural increase (r), and the carrying capacity of the environment for the blue-fronted parrot population (K) (fledglings/km²). Six methods were used to estimate the intrinsic rate of natural increase, and five methods of density estimation were used as indicators of carrying capacity.
5. Previous to their estimation, age-specific survivorship and reproductive curves had to be produced. In the absence of this information from the wild or from captivity, eight different survivorship curves were used, to later disregard some of them, putting more emphasis on the most plausible ones. Also several reproductive curves were developed, depending on demographic traits such as age of first reproduction and “skipping rates” (intermittence of reproduction).

6. The estimates of the intrinsic rate of natural increase (r) were: 0.12-0.18 with the Lotka equation, 0.31-0.32 with Cole's formula, 0.15-0.20 with the generation time and replacement rate method, 0.3 with the homeotherm meta-analysis regression, 0.34-0.46 with a two ages projection matrix, and 0.23-0.28 with Lande's equation.
7. Other demographic features were produced as "secondary" estimates: average longevity, finite reproductive rate, stable age distribution, relative proportion of fledglings and adults, and proportion of non-breeding females in the adult female population. These estimates are called "secondary" because they are based upon the estimation of other demographic parameters and not upon independent biological data.
8. The carrying capacity estimates were approximated assuming that present densities can be considered as a minimum value of K , unless the blue-fronted parrot population is already at its K value, or in a very favorable year. In the "*Loro Hablador*" Natural Reserve, using a direct density method (based upon a surface area of 250 m on each side of transects or paths), the fledgling density at the time of the start of the harvest period was 3.6 fledglings/km² (lower and upper 95% CL= 0.3 and 6.8), and 2.6 fledglings/km² (lower and upper 95% CL= 0 and 5.8), for the 2002-2003 and 2003-2004 reproductive seasons, respectively. For the same periods the estimates were 12.7 and 17.1 fledglings/km² with the transect method, 1.9 and 4.4 fledglings/km² with the Byth and Ripley distance method, 4.6 and 4.5 fledglings/km² with the T-square distance method, and 4.1 and 6.6 fledglings/km² with the adaptive sampling method. Additionally the fledgling density was estimated from a special survey of nest-trees carried out in the management area of the "*Impenetrable*" in July 2004; applying a direct method of density estimation (drawing a series of trapezoids around all nest-trees and houses assigned an offtake quota in the 2004 season) the density in the management area was 26.6 fledglings/km².
9. The Byth and Ripley distance method was the least reliable of all, for due to the spatial sampling distribution very few quadrats were feasible ($N= 6$ and 5 quadrats for the 2002-2003 and 2003-2004 reproductive seasons, respectively). The transect and the adaptive sampling methods were considered the most reliable ones, the former because it used the Fourier series distribution to avoid depending upon the spatial pattern of the population, and the latter because it considers the population as following a contagious dispersion pattern. The fledgling density estimation based upon the survey of nest-trees carried out in the management area of the "*Impenetrable*" in July 2004 is probably an overestimation because it considered only the dense patches of nest-trees attractive to the collectors for the offtake. All these estimates were calculated with their 95% confidence limits; the upper 95% confidence limit of the fledgling density from all methods oscillated between 6.2 and 46 fledglings/km² (not considering the Byth and Ripley distance method). Being actual average densities a potential minimum estimate of the carrying capacity, it was considered that the average carrying capacity of the

“*Impenetrable*” for the blue-fronted parrot should be between 2 and 26 fledglings/km² on an “average” year.

10. In an effort to narrow down these wide ranges of parameter estimates, data from seven years of offtake (seasons 1978/1979 to 2003/2004) was used. The offtake remained relatively constant at an average of 1.47 fledglings/km² ($n=7$, std. dev.= 0.24, lower 95% CL= 1.29 and upper 95% CL= 1.65). This suggests that the offtake seems to be maintained (and even seems to increase slightly in the last four years) at a level that apparently can be sustained by the blue-fronted parrot population. I resorted to three scenarios: (a) that the population density has remained stable, (b) that the population density has declined 20% in those seven years, and (c) that the population density has increased 20% in those seven years. Then I asked the question: which are the r and K values compatible with the above assumptions (a)-(c) in the presence of the an average annual offtake of 1.47 fledglings/km²? Using possible field fledgling densities between 4 and 12 fledglings/km², the Solver procedure of the Excel spreadsheet was used to answer this question, and the parameter values were narrowed down to an r value of around 0.3 and K values between 18 and 22 fledglings/km², if the blue-fronted parrot population were in a stable state; r values slightly above 0.3 and K values between 14 and 18 fledglings/km², if the blue-fronted parrot population were decreasing 20% in seven years; and r values slightly below 0.3 and K values between 22 and 26 fledglings/km², if the blue-fronted parrot population were increasing 20% in seven years.
11. Nevertheless, to be conservative and to have a complete picture of the behavior of the logistic model of population growth with the fixed escapement management rule was run with lower minimum r and K parameter values: an intrinsic rate of natural increase (r) between 0.1 and 0.6 (with steps on 0.1), and with carrying capacities (K) between 2 and 26 fledglings/km² (with steps of 2 fledglings/km²). From all the possible 138 combinations, three of them ($r=0.2, 0.3$ and 0.4 , and $K=20$ fledglings/km²), were considered as the most plausible ones, and the pair $r=0.3$ and $K=20$ fledglings/km² was considered the most credible one.
12. The logistic model of population growth included indirectly some environmental stochasticity. It was implemented by affecting the carrying capacity with a random variation that depended on an artificially assigned coefficient of variation of K , assuming that K has a normal distribution. Three values of the coefficient of variation of K used were: 0, 20 and 40 %.
13. The logistic population model also included the “*fixed escapement*” rule of population management. This rule was selected because it represents one option of the so-called “feedback” management rules, and because it is considered as very efficient in coping with uncertainties of various origins: climatic, biological, or social (e.g., poaching). The model was programmed so that the optimal escapement density could be selected by maximizing the cumulated offtake of

- fledglings during 30-years of simulation. This was done for the three levels of stochasticity of the carrying capacity (K).
14. The optimal sustainable escapement (threshold) densities were calculated for all selected values of the r and K parameters, and resulted in values that had a range between 0.6 and 10 fledglings/km², with a small tendency of being lower as the stochasticity level of K was higher. It was observed that K had a larger influence than r in determining the escapement density. Using $K=20$ fledglings/km², the optimal sustainable escapement (threshold) density for blue-fronted parrot fledglings were 6.1, 7.5, and 7.8 fledglings/km², for $r=0.2, 0.3$, and 0.4 , respectively. That is, any excess over those three densities could be harvested, depending upon the r value assigned to the blue-fronted parrot.
 15. The optimal sustainable offtake densities were also calculated for all selected values of the r and K parameters, and resulted in values that had a range between 0.04 and 3.84 fledglings/km², also with a small tendency of being lower as the stochasticity level of K was higher. Using $K=20$ fledglings/km², the optimal sustainable offtake densities for blue-fronted parrot fledglings were 0.8, 1.45, and 1.96 fledglings/km², for $r=0.2, 0.3$, and 0.4 , respectively. The most credible combination of r (0.3) and K (20 fledglings/km²) resulted in an optimal sustainable offtake density of 1.45 fledglings/km², quite close to the seven-years field average of 1.47 fledglings/km².
 16. A discussion is presented over several weaknesses of the methods used, some comparisons are made with the results from the literature, and some general considerations are given about the blue-fronted parrot management. It is also emphasized that these results, based upon the logistic model of population growth and the fixed escapement density management rule, represent an *optimal* sustainable offtake, but that many other levels of offtake may also be sustainable.

1 Introduction

A management plan for a renewable natural resource cannot be a recipe: management plans constitute such a complex decision making process, that usually they have to be applied in an artisanal way, even when based upon the most rigorous scientific methodology. In the case of the blue-fronted parrot (*Amazona aestiva*), the results here presented are essentially valid for the Dry Chaco region, and particularly to the so called “*Impenetrable*” of the Chaco Province (Argentina), and whatever management plan is devised it cannot be automatically extrapolated to the other areas of the Chaco region where the blue-fronted parrot is being harvested (Province of Formosa), and even less for the Yungas rainforests in the Provinces of Salta and Jujuy. A model for those habitats still will have to be developed for a more comprehensive sustainable management plan of *A. aestiva*. This is particularly important because since 1981 *A. aestiva* has been included on Appendix II of the Convention on International Trade of Endangered Species (CITES) as a species for which actions to regulate trade and over-exploitation are needed. The need for a management plan based on a model that would help management decisions has already been stressed by Beissinger and Bucher (1992).

The “Blue-fronted Parrot Project” (known as “*Proyecto Elé*”) is being carried out by the Secretary of the Environment and Sustainable Development of Argentina, in partnership with the Provinces of Chaco, Formosa, Jujuy and Salta, and with a relationship with the Provinces of Córdoba, Corrientes, Santa Fe, Santiago del Estero and Tucumán. The management model here proposed is based upon a management rule that is well adapted to the conditions of “*Proyecto Elé*”. The management rule used is the “*fixed escapement*” rule, also known as the “*Bang-Bang*” rule. It has been accepted that this rule is a very efficient in incorporating environmental and demographic stochasticities (Ludwig and Walters, 1981; Walters, 1981; Hilborn y Walters, 1992, Lande *et al.*, 2001); however, it poses a strong demand on management logistics: it requires that each year a population sampling or a survey be carried out before defining the level of the harvest. Thus no management plan based upon the “*fixed escapement*” rule can be practical unless an efficient sampling method is available from the standpoint of the statistics, the economics, and the logistics. The “*fixed escapement*” management rule also belongs to the group of rules called “state-dependent” rules; because the dynamics at a given time depends upon the state of the system at the previous time, the state of the population (and of its environment) has to be determined prior to the harvest and, to be effective, this has to be done in a rigorous, cheap and simple way.

Additionally, the “*fixed escapement*” management rule requires that the “*escapement*” population level be determined. This level is defined as the population threshold below which no harvest should take place. Sometimes this threshold may be known when a given wildlife species has been managed in the field for some time, and the population response to harvest has been recorded and analyzed. This is not the case of the blue-fronted parrot management, so more theoretical approaches for determining the “*escapement*” population level had to be used. These, in turn, require a relatively good knowledge of the population dynamics of the managed species, as well as a satisfactory

estimate of the main population dynamics parameters, in order to develop reliable models to be used in the determination of the “*escapement*” population threshold.

Information on the the blue-fronted parrot population dynamics and on its parameter values is very limited. For this reason, in order to develop a model to determine the “*escapement*” population level, I decided to carry out the modeling with two strong restrictions: (a) to develop the model exclusively for the “*Impenetrable*” area of the Province of Chaco, and (b) to use the logistic population dynamics model of population growth. There are two reasons for the first restriction: it is the area where the best biological and ecological information on *A. aestiva* is presently available under wild conditions in Argentina, and it is also the region where –relatively speaking– the offtake pressure is higher. The justification for the second restriction is based on three facts: (i) the logistic model does not impose strong demands on the biological and ecological information needed, (ii) it has been widely used in wildlife management, and (iii) it has been shown to have many adequate and satisfactory properties for this kind of applications (Runge and Johnson, 2002).

The “*fixed escapement*” management rule is also a form of applying the adaptive management approach, that is being increasingly used as a standard in wildlife management (Walters 1986, Lancia *et al.* 1996).

In this report I apply a population dynamics logistic model to the blue-fronted parrot population, and using a “*fixed escapement*” management rule I simulate the harvest of fledglings¹ to determine the optimal level of sustainable use for the case of the “*Impenetrable*” region of Argentina. Before describing the methods, I summarize below the main characteristics of the study site and of the present management program.

2 Study site and present management program

2.1 The region

Argentina participates with around 50% of the surface of the American Chaco, which occupies 1.1 million of km², also involving extensive territories of Bolivia and Paraguay, and a very small portion of Brazil. After the Amazonia the American Chaco is the biggest forested area in the South American continent. The Chaco Region of Argentina embraces about 600,000 km², and it occupies 22% of the continental surface of the country. It constitutes its biggest forested region, involving ten Provinces: Formosa, Chaco and Santiago del Estero in its totality, and substantial parts of the north of Santa Fe and San Luis, the east of Salta, Tucumán, Catamarca and La Rioja, and of the north and west of Córdoba. Topographically it constitutes a vast plain with a soft slope toward the east.

¹ I will be calling fledglings those parrots that are present in the nest at the start of the offtake season. Strictly speaking they might still be called chicks, but at that time they are almost ready to fly. Actually many of them start at that time to leave the nest, rehearsing short flying trips, and return to the nest. This happens in the study area of the “*Impenetrable*”, where the offtake is the latest within the harvest season in the Argentinean Dry Chaco. Calling the collected birds fledglings is done only for convenience, and it would not be valid in northern areas of the Argentinean Dry Chaco, where offtake starts earlier.

The climate is subtropical warm, with record absolute maximum temperatures for the continent. Although with a marked climatic gradient, it shows a mean annual temperature between 18 and 26°C, high potential evapotranspiration (between 900 mm in the South and 1.500 mm in the North), and an annual maximum precipitation of 1.300 mm (near to the Paraguay river), although in most of the Dry Chaco precipitations are of the order of 500 mm/year, reaching minimum values of 300 mm/year in arid pockets. In places of scarce forest cover soils receive extremely high caloric energy, that generates water deficits for plants as well as nitrogen and carbon deficits due to volatilization (the soil organic matter literally burns).

The region presents great diversity of environments: extensive plains, “*sierras*”, big rivers, dry and flooded savannas, wetlands, saltpeters, and a great extension and diversity of forests and shrubs. However, the semi-arid and arid Chaco (where this work was carried out) is dominated by xerophilic forests (identified with the common name of the “*Impenetrable*”), and depending upon the biogeographical district, also by savannas and grasslands. The surface of the Chaco Forest remaining is of about 230,000 km². Fig. 1 shows the range of *A. aestiva* in South America, as well as its distribution in Argentina.

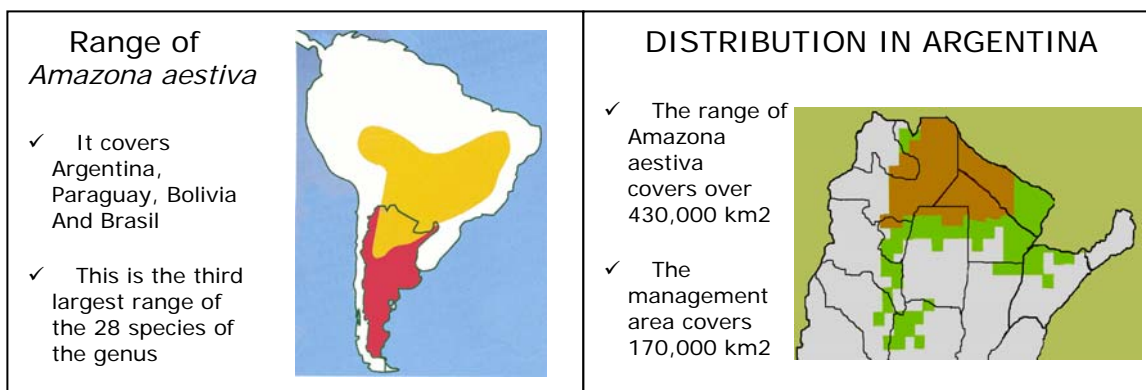


Fig. 1. Range of *A. aestiva* in South America and in Argentina. In the map of the right side, the northernmost darker shade corresponds to the area under management.

2.2 Use of the blue-fronted parrot in Argentina

For nine years (1983-1991) around half a million live blue-fronted parrots were authorized by Argentinean provinces to be used in the national and international pet trade. This offtake peaked in 1985, with about 75,000 blue-fronted parrot permits granted. Authorities decided then to reduce the number of permits to about one half of the historical average, which is shown as the irregular decrease after the 1985 peak (Fig. 2). However, national Argentine authorities considered even this to be excessive and unsustainable, and a four-years total ban was declared from 1992 to 1995, that was used to gather some basic biological and ecological information about the blue-fronted parrot and its habitat.

In addition to being excessive, this offtake was carried out within a context that made the harvest of the blue-fronted parrot not only biologically but also administratively and socially unsustainable. Some of the most notorious problems of the harvest between 1983 and 1991 were: (a) quotas were based upon trade criteria and not harvest criteria, (b) there was an extremely unequal distribution of the wealth created by the parrot trade, with local people receiving an insignificant fraction of the value of the parrots in trade, (c) there was no uniform policy among provinces involved in permit granting, (d) there was a low level of control with high uncertainty about the real origin of specimens, (e) the offtake was carried out with a systematic destruction of the nest-trees, (f) as the blue-fronted parrot is considered a pest of citrus plantations, they were being killed in high numbers, and (g) a high mortality usually took place during stockpiling and transport.

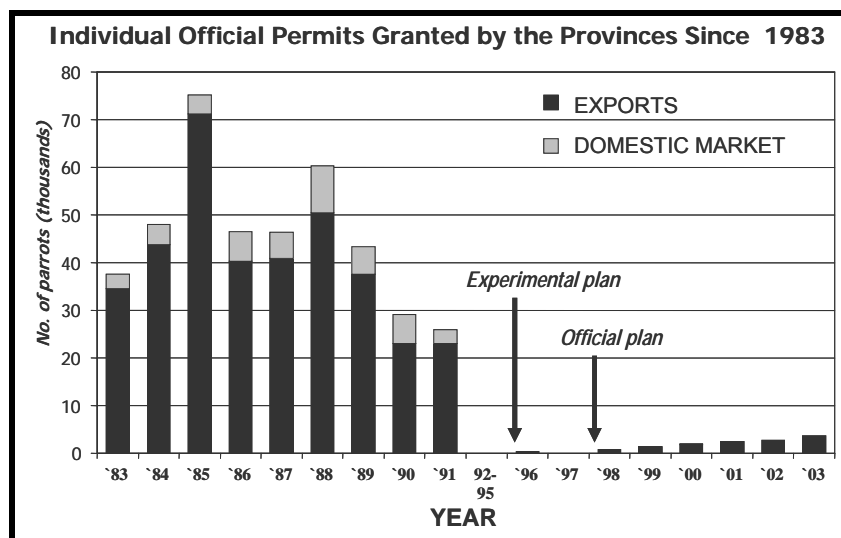


Fig. 2. Provincial official permits of blue-fronted parrot trade, for export and domestic trade.

In 1996 an experimental management plan was initiated, providing an experience that was used to later devise a sustainable management plan that is in effect since 1998 to present, designed in such a way as to avoid the pitfalls described above (a-g).

Presently each year a team of around 40-50 people works in the field for the “*Proyecto Elé*” during the offtake period of fledglings in the Dry Chaco (roughly between December and February). This is an expensive operation, and the cost is being covered by the program itself. This was made possible because every blue-fronted parrot being exported implies a certain amount of money collected from the exporters, which is deposited in a trusteeship fund. Additionally the local collectors obtain an amount of money that, in relative terms to the final exported price, is between three and five times larger than the one obtained during the previous management system. In absolute terms and in United States dollars, local people receive between five and ten more income than with the previous management system. The new management plan requires that all local collectors be land owners, as a way of stimulating local families to acquire the land they work on, to secure a future extra annual income form the blue-fronted parrots, and incentivating a sustainable use mentality. In the management area, after harvest, every

nest has to be left with at least one fledgling. Under this scheme of management the “*Proyecto Elé*” hopes to keep the offtake within determined levels, so that the use of *A. aestiva* can be considered sustainable from the biological-ecological as well as socio-economic point of view. Additionally, with the trusteeship fund’s resources a 235 km² Blue-Fronted Parrot Reserve (“*Loro Hablador*” Natural Reserve) was created in the “*Impenetrable*”, where not only biological and ecological information is being collected, but also to possibly serve as a population “source” that can replenish harvested areas.

Fig. 3 shows the offtake levels for the last seven years under the present management scheme. It is interesting to observe that the blue-fronted parrot fledgling offtake remained below the allowable quota. The total offtake of fledglings has been increasing since 1998, but as a result of extending the area subjected to harvest (that is, including more local people receiving harvest quotas): the geographical area is extended, but the offtake quota in each harvested unit remained relatively constant.

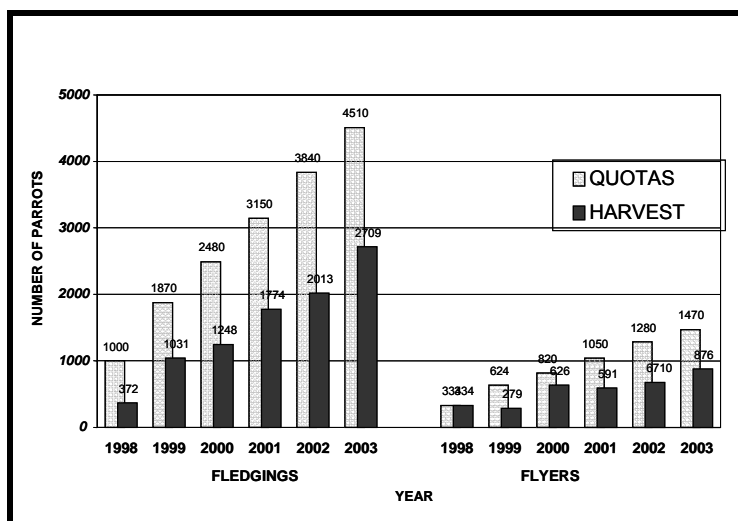


Fig. 3. Offtake levels and allowable quotas from 1998 to 2003 under the present management scheme, for both fledglings and flyer parrots.

Fig. 4 depicts the geographical distribution of fledgling collection centers under the present management scheme (not in scale). The triangular shaped area in the Chaco Province (with many small properties and one property larger than 30 km²) represents the area of about 20,000 km² where this pilot modeling study was carried out. This area represents about 12% of the region under management.

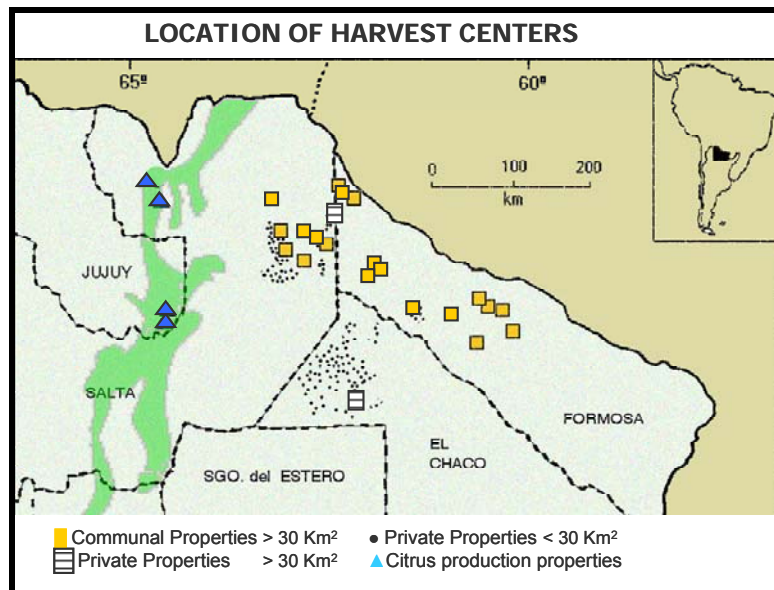


Fig. 4. Geographical distribution of fledgling collection centers under the present management scheme (not is scale).

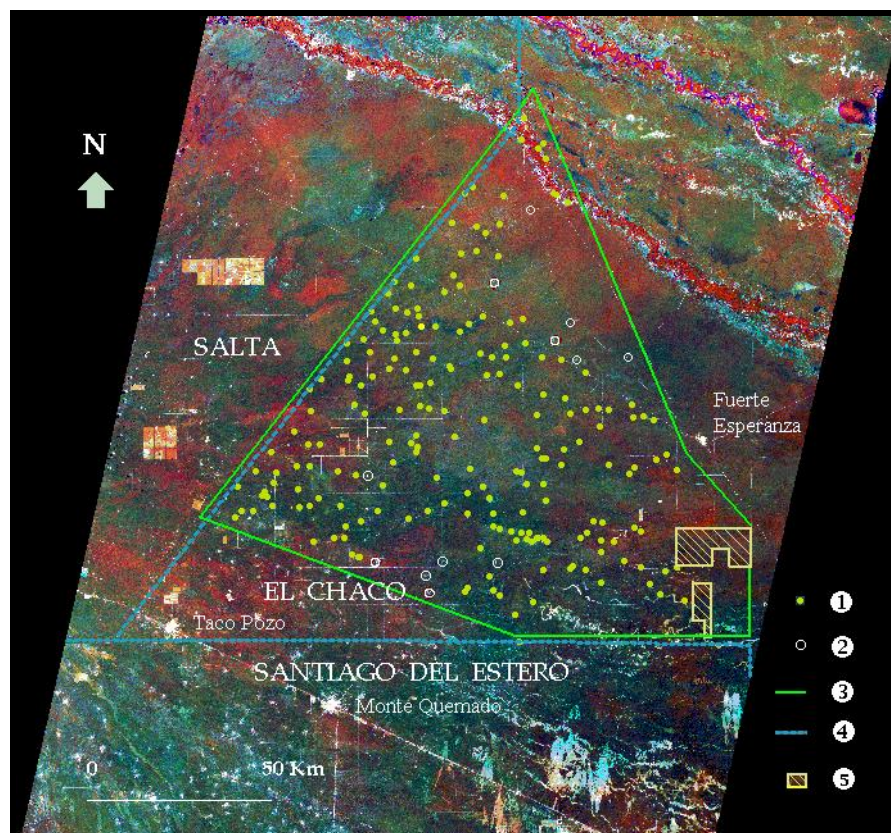


Fig. 5. Satellite image of the “Impenetrable” of the Dry Chaco of the area used for analysis and modeling. See text for symbol interpretation.

Fig. 5 shows a satellite image of the area covered in the present analysis. The approximately triangular shape represents the area of the “Impenetrable” subjected to the

modeling here carried out. In Fig. 5 the small green dots (symbol ①) represent the localities with the households that received a quota; the white hollow circles (symbol ②) are the households that have adequate habitat for the blue-fronted parrot but that have not been assigned a quota; the green line (symbol ③) shows the limits of the area of the “Impenetrable” subjected to analysis in this modeling; the blue line (symbol ④) shows the limits between the provinces of Salta, Chaco and Santiago del Estero for the Dry Chaco; and the area in yellow in the lower right corner represents the “*Loro Hablador* Natural Reserve”(symbol ⑤). Different colors represent not only the effects of different land uses, but also the effects of an extremely variable precipitation pattern, showing different colors even within a homogenous habitat composition and structure.

3 Methods

3.1 Parameter estimation

The logistic population dynamics model of population growth, in its most basic form, depends only upon two parameters: the intrinsic rate of natural increase (r), and the carrying capacity of the environment for the blue-fronted parrot population (K). I will deal with the estimation of each of them separately.

3.1.1 Estimation of the intrinsic rate of natural increase (r)

As the biological-ecological information on the blue-fronted parrot is very scarce, I resorted to a variety of methods for the estimation of r , to try to verify the robustness of the estimate of this parameter. I used six different methods: (i) the Lotka equation, (ii) Cole’s formula, (iii) an estimate based upon the replacement rate and the generation time, (iv) a two ages matrix model, (v) an homeotherm meta-analysis, and (vi) the Lande equation.

3.1.1.1 The Lotka equation

One of the most common estimation methods of r uses the Lotka equation (Lotka, 1922), also known as Euler equation:

$$\sum_{x=\alpha}^{\omega} e^{-rx} l_x m_x = 1,$$

where x is age (years), l_x is the survivorship age-specific schedule (probability of being alive at year x), m_x is the reproduction age-specific schedule (also called the maternal function), and represents the number of female offspring produced by an average female aged x years, and the summation is carried out from the first (α) to the last (ω) reproductive age. There is no explicit solution for r , so the Solver procedure of MicroSoft Excel spreadsheet was used.

There are no complete estimates of l_x for the blue-fronted parrot, but only for the survival from egg laying to fledgling. From the first age (juveniles of year 1) onwards seven survival curves were used: four of them are the classical ideal survival curves usually

called curves type I, II, III and IV (Deevey, 1947; Hutchinson and Deevey, 1949); type I is the *negatively skewed rectangular* curve (most of the members of a cohort die more or less simultaneously at an age considered to be characteristic of the species); type II is the *diagonal* curve, and implies a constant mortality rate for all ages; type III implies a constant number of individuals dying for all ages, and was generated as intermediate between curves type II and type IV; type IV is the *positively skewed rectangular* curve (extremely heavy mortality beginning early in life but the few individuals which survive to advanced ages have a relatively high expectation of further life); type I-II, was generated as laying between type I and type II curves; finally two sigmoid-shaped curves were also used, one that starts with survival values above (Sig-A) and another one that starts with survival values below (Sig-B) the *diagonal* curve (Fig. 6). Few complete survivorship curves have been developed for birds, but Deevey (1947) and Lack (1966) claim that most bird species have survivorship curves either type II or intermediate between type I and type II (for example, the whooping crane *Grus americana*; Binkley and Miller, 1980), and sometimes a sigmoid shaped survivorship curve (as in the case of the American herring gull, *Larus argentatus*; Marshall, 1947). Five species of birds reared in Zoos also showed survivorship curves intermediate between type I and type II (Comfort, 1962, Ricklefs, 1998).

As the form of survivorship curves of bird species are either very general (between survivorship curves type I and II) or too specific (several intermediate survivorship curve types) (see Deevey, 1947, Lack, 1966, Marshall, 1947, Rowley and Chapman, 1991), I also applied Binkley and Miller's (1980) method to get an approximate idea of the possible form of the survivorship curve of the blue-fronted parrot. This method was designed by Binkley and Miller (1980) to obtain the survivorship curve and the maximum longevity of the whooping crane *G. americana*, a species for which no age structure was known (except the young of the year), nor the sex was determined. This situation is similar to the one of the blue-fronted parrot, except that population studies of the whooping crane provided a long series of censuses with the values of population size for several years. Binkley and Miller (1980) propose the use of the relationship

$$N(t) - y(t) = \sum_{x=1}^T l_x y(t-x)$$

Where $N(t)$ is the total population size in year t , $y(t)$ is the number of young in year t , x is the age in years, and T is the maximum age (that is, $x = T$: $l_x = 0$). This means that the number of adults in year t equals the total number of young of past years surviving to that year.

To apply Binkley and Miller's (1980) method to the blue-fronted parrot, as there are no prolonged censuses available, I had to resort to several assumptions: (a) I assumed that the total population remains constant, (b) the value of the total population was based upon the estimate of the number of nest-trees (see section 3.1.2.) multiplied by a factor; this factor assumed the value 3, because there are two parent parrots per nest –male and female– and it was supposed that some individuals do not breed (either because they “skip” a breeding season –see “skipping rate” below– or because a fraction of the

population delays the initiation of reproduction until they reach a certain age –see α or age of first reproduction below–), and (c) the proportion of fledglings among the total population also remained constant (i.e., I am assuming a stable age distribution).

Based upon these three assumptions a spreadsheet was prepared, and the following procedure was applied: (i) a second degree polynomial (as suggested by Binkley and Miller, 1980) was used to calculate mortality rates (d_x), (ii) mortality rates were converted to absolute mortality (number of deaths) as applied to the total number of fledglings (this stage was considered the initial population, because the l_x series to be calculated estimates only the survival for juveniles one-year old onwards), (iii) number of deaths were converted to l_x , and (iv) the l_x values were converted into number of survivors. I used MicroSoft Excel's Solver utility to find the three coefficients of the second degree polynomial such that the calculated mortality rates generated an l_x curve that produced an accumulated number of survivors for 30 years equal to the total adult population size. The Solver procedure was implemented under three constraints: (a) $l_x > 0$ for all $x < 30$, (b) $l_x = 0$ for $x = 30$, and (c) $d_x \geq 0$ for all x . The resulting survivorship curve was identified as BM-80.

To be able to manipulate survival curves mathematically, whenever possible the survival curves were fitted to a Weibull function, which is one of the most common functions to describe survival curves (Ricklefs, 1998, Runge and Johnson, 2002; Chaves *et al.*, 2004); if fit was not satisfactory a polynomial function was used. The Weibull function can be expressed to fit either mortality or survival; I preferred the latter, and the following form of the Weibull function was used:

$$l_x = e^{-(\lambda x)^\rho}$$

where, as before, x is the age in years, and λ and ρ are the scale and the form parameters of the Weibull function, respectively. The exceptions to the Weibull function fit were the survivorship curve type II, that being a straight line was fitted by simple linear regression, and the survivorship curves SIG-B and BM-80, that were fitted to a second and third degree polynomial, respectively. Once a mathematical expression of the age-survival schedule was established the average longevity was calculated by simply averaging the number of years lived by the survivors from one year to the next.

A critical aspect of any survivorship curve is the maximum life span used (Stearns, 1992). A maximum average life span of 30 years was used because it was considered reasonable by some experts (Igor Berkunsky, *pers. comm.*), by some informal appraisals (see Appendix V), as well as by an aviary owner that rears *A. aestiva* in Europe (E. Fernández Nogales –see Appendix VI–). Another indication of the maximum life span was obtained by regressing weight, reach and longevity data from a variety of sources. On one hand I used a linear regression of weight (g) on reach (cm) based upon the data of eight Psittacidae species (see Appendix I); as these eight species belong to only three genera, some sort of "comparative method" technique would have been appropriate to correct for the possible effect of a common ancestor (e.g., the phylogenetically independent contrast method); however this would require a good knowledge of the

phylogenetics of these species of Psittacidae, so a simple linear regression was used and no correction for the possible effect of a common ancestor was applied. Additionally I used a double logarithm linear regression based upon the data of 363 species of various of families; finally I calculated another linear regression of reach (cm) on *average* longevity (years) based upon the data of 27 species of various families. The details of the data and species used, as well as the source of these data are given in Appendices I, II, III and IV.

Once a given age-specific survival schedule has been selected, an age-specific fecundity schedule has to be used in order to use Lotka's equation to estimate r . There is no information in the literature on the age-specific fecundity of the blue-fronted parrots. Information from the "*Loro Hablador*" Natural Reserve provided by Igor Berkunsky (*pers. comm.*) from the 2002/2003 (from now onward called 2002) and 2003/2004 (from now onward called 2003) field research campaign was used. This information included density of nests, average number of eggs per nest, and survival from egg to fledgling. Apparently *A. aestiva* is a continuous breeder in captivity, but it may be an intermittent breeder in the wild; however, no data on the "skipping rate" (the number of non-reproductive years in between reproductive years) is available, so several "skipping rates" (0, 1, and 2) were applied to the age-specific fecundity schedule and tested in estimating r . This definition of "skipping rate" should not be confused with the formal definition given by Hunter *et al.* (2000), which stands for "proportion of birds known to have bred the year before that did not attempt to breed in the current year".

The age of first reproduction (α) is considered an important demographic parameter because the intrinsic rate of natural increase (r) is usually quite sensitive to the value of α (Cole, 1954). The "best guess" of biologists familiar with *A. aestiva* suggested α values of the order of one and two years, or at the most, three years (Ricardo Banchs and Igor Berkunsky, *pers. comm.*). However, values of α of 4 and 6 years (and even higher) emerged from the experience in some blue-fronted parrot aviaries (see Appendix VI). I resorted to the study of Sæther and Bakke (2000), who provided a detailed Appendix with four demographic parameters for 49 species of birds, including the age of first reproduction. I carried out a multiple linear regression with the age of first reproduction as dependent variable, and the other three demographic parameters given by Sæther and Bakke (2000) as independent variables: the fecundity rate (the number of female offspring produced at the end of the nestling period per adult female per season), the juvenile survival rate (the annual survival rate during the first year of life), and the adult annual survival rate (for data, species and sources see Ecological Archives E081-005-A1 in <http://www.esa.org>). This is considered to be a crude procedure because no other possible optimal relationship was explored nor a correction for phylogenetic factors was contemplated; the reason is that the main objective was only to be able to have a rough approximation for the age of first reproduction of the blue-fronted parrot based upon the demographic parameters of other bird species.

The progeny sex ratio is also a critical variable in the application of Lotka's equation for estimating r , because all calculations are carried out as if the whole population would be a "virtual" exclusively female population, with the assumption that there will always be enough males available to fertilize all females. I used a 1:1 progeny sex ratio (i.e., 50%

males and 50% females) despite very little information is available, either under wild or commercial rearing conditions. In aviaries, under extremely favorable feeding and rearing conditions, the progeny sex ratio of *A. aestiva* seems to be strongly male-biased (up to 30 males:70 females ratios; Eugeni Fernández Nogales, *pers. comm.*; see Appendix VI). However, the literature on progeny sex ratio of birds (see Appendix VII for a discussion of some recent findings) shows that changes in progeny sex ratio in relation to ecological and behavioral factors is not decisive, despite the general acceptance that there must be an adaptive adjustment in offspring sex ratio by females as predicted by evolutionary theory. However the few field determinations of offspring sex ratio in the parrot family tend to show that apparently there is no alteration in the 1:1 offspring sex ratio. South and Wright (2002), working with the Yellow-naped Amazon (*A. auropalliata*), determined that the sex ratio of nestlings was 51% male (39 out of 77), which did not represent a statistically significant departure from unity; they also found that the mean proportion of males per brood ($n = 37$) was 49%, which did not differ from unity, nor did the sex ratio of first-born or second-born nestlings. In a limited experience with field *A. aestiva* populations Fernandes Seixas and Miranda Mourão (2002) found that the sex ratio for a sample of 27 chicks from 15 nests during the reproductive season was not differing significantly from 1:1. Heinsohn and Legge (2003) have confirmed that the parrot *Eclectus roratus* has a strongly male-biased sex ratio in nature (1.47 in favor of males) but they also claim that this species has an even sex ratio at fledging, and suggest that there must be a higher mortality of adult females after hatching. More justifications on why I used a 1:1 progeny sex ratio for the blue-fronted parrot can be found in Appendix VII.

3.1.1.2 Cole's formula

Cole (1954), provided a formula to estimate r (known as Cole's formula):

$$\exp(-r) + b \exp(-r\alpha) - b \exp(-r(\omega - \alpha + 1)) = 1$$

where r is the intrinsic rate of natural increase, α is the age of first reproduction, b is the clutch size (number of eggs laid per female per year), and ω is the age of last reproduction. As, with the exception of b , these parameters do not have a reliable field estimate, several plausible values were used for α (1, 2, 4 and 6 years) and for ω (5, 10, 15, 20, 25 and 30 years). Additionally, although the value of b is relatively well known (average of about 4 eggs per female per year) I used the values of 2, 3, and 4 eggs per pair per year; these values were divided by two because Cole's formula is expressed exclusively on the female fraction of the population (using a sex ratio of 1:1; see explanations in section 3.1.1.1). There is no explicit solution to estimate r from Cole's equation, so the estimate was carried out using the Solver procedure of MicroSoft Excel spreadsheet.

3.1.1.3 Use of the replacement rate and the generation time

Both the replacement rate (also known as the net finite reproductive rate) and the generation time are “secondary” parameters, in the sense that they are derived from the age-specific survival and fecundity schedules (l_x and m_x , respectively).

The replacement rate is identified as R_o and is defined by $\sum l_x m_x$; it represents the actual number of offspring an average individual produces in one lifetime, taking into consideration probabilities of age-specific death as well as age-specific reproductive rate for the parent.

On the other hand the generation time, also known as cohort generation time, is identified as T_c and is defined by $(\sum x l_x m_x) / R_o$; it represents the average length of time between birth of a parent and birth of its middle progeny. It can also be interpreted as the average age at which, if an individual would reproduce only once (leaving the same progeny as in multiple reproductive efforts), the rate of growth would be the same.

Once R_o and T_c have been calculated, the intrinsic rate of natural increase (r) is defined by (Krebs, 1989):

$$r = (\ln R_o) / T_c$$

3.1.1.4 Two ages matrix approach

Although there is no information on the age structure of the blue-fronted parrot population a two ages matrix approach, as the one used by Hiraldo *et al.* (1996) for the lesser kestrel (*Falco naumanni*), was attempted, for it requires only some parameters that (as developed in the previous sections) can be estimated, or at least assigned a range of plausible values. The method to estimate the intrinsic rate of natural increase (r) and its variance can be found in Lande (1988), and the one to calculate the absolute sensitivity for the various demographic parameters in Kendall and Stuart (1977) or the elasticities (relative sensitivity) in Kroon *et al.* (1986, 2000) and Caswell (1989).

Following Hiraldo *et al.* (1996) the population is assumed to be composed of two age classes: up to one year olds (fledglings or yearlings: Y), and adults (A). The following parameters are considered: b = female progeny that fledged per reproductive female, c_o = proportion of yearlings attempting breeding, c_a = proportion of adults (2 years old or older) attempting breeding, s_o = juvenile survival (i.e., survival during the first year of life), and s = adult annual survival.

With the demographic parameters defined above the model used by Hiraldo *et al.* (1996) takes the form (in matrix notation):

$$\begin{pmatrix} Y \\ A \end{pmatrix}_{t+1} = \begin{pmatrix} c_o b s_o & c_b s_o \\ s & s \end{pmatrix} \times \begin{pmatrix} Y \\ A \end{pmatrix}_t$$

where the four cell matrix is called the projection or transition matrix and denoted as $\mathbf{M}(t)$, the two cell vectors represent the number of individuals in each age class and denoted as $\mathbf{V}(t)$, where t refers to time.

The basic characteristic equation of $\mathbf{M}(t)$ is given by $[\lambda^2 - s\lambda - s_0sb = 0]$ and has two solutions; the dominant, real-valued solution (λ_1) is an estimate of the annual rate of change of the population. If $\lambda > 1.0$, the vital rates suggest the population is increasing; if $\lambda = 1.0$, the population size is stable; and if $\lambda < 1.0$, the population is declining. We have to be cautious in using this estimate of λ based on matrix models to forecast future population sizes for it is only an estimate of how the population was changing over the sampling period; i.e., we could use this estimate of λ to project future population size only under the assumption that current estimates of the vital rates remain constant. In contrast with the logistic model, in the population dynamics as governed by matrix models (also called Leslie or Lefkovitch models) matrices grow, or decline, exponentially (except for the exceptional case where $\lambda = 1.0$). This matrix model is clearly unrealistic for the long-term growth or decline of any natural population. There exist, however, much more realistic density-dependent matrix models.

The latent root of the largest absolute value (which is designed as λ_1 , and is the dominant eigenvalue of the matrix), is equivalent to e^r . A population of arbitrary structure that changes accordingly to the values of $\mathbf{M}(t)$ will, if the matrix is square, nonnegative and irreducible, approach an age vector called the “stable age distribution”. At this point each age class will be increasing by λ_1 times during each time interval, i.e., age distribution stabilizes, and population structure at any future time can be predicted. Sensitivities and elasticities of the demographic parameters as well their contribution to r (Kroon *et al.* 1986, 2000, Caswell, 1978), were programmed and processed with the MATLAB software.

The dominant latent root of the matrix $\mathbf{M}(t)$, λ_1 , is most usually determined by taking powers of the matrix. When $\mathbf{M}(t)$ is raised to a sufficiently high power k , the matrix \mathbf{M}^k stabilizes, in the sense that when \mathbf{M}^{k+1} is divided by \mathbf{M}^k , element-by-element, the result is a matrix of identical ratios that does not change for higher powers of \mathbf{M} (Keyfitz, 1968). The common ratio is λ_1 . The stable age distribution is obtained, then, by dividing each element of \mathbf{M}^k by λ_1 . This operation produces a matrix \mathbf{Z}_1 . Each of the columns in the \mathbf{Z}_1 matrix is proportional to the stable size distribution (the population size structure that will be achieved if any initial population distribution is allowed to change by the set of transition probabilities $\mathbf{M}(t)$ for an indefinite period of time).

3.1.1.5 Homeotherm meta-analysis

Fenchel (1974) carried out an analysis of the relationship between animal mass (weight) and the intrinsic rate of natural increase (r). He carried out the analysis for unicellular, heterotherm and homeotherm organisms. Although Fenchel (1974) recognized that the data for the homeotherm animals was too scanty to be reliable, he suggested a pair of values for the coefficients of a regression for this group:

$$r = aW^n$$

where r is the intrinsic rate of natural increase and W is weight in kg. The values of the regression coefficients suggested by Fenchel (1974) were $a = -1.4$, and $n = -0.275$.

3.1.1.6 Lande's equation

Lande (1988) proposed the following relationship between five demographic parameters:

$$\lambda^\alpha \left(1 - \frac{s}{\lambda}\right) = (l_x)b$$

where λ is the annual geometric (or finite) growth rate (i.e., $\exp(r)$), s is the annual adult female survival, α is the female's age at first breeding, l_x is the probability of a fledgling surviving to α , and b is the productivity per female per year. Most of these parameters have been discussed in relation to the previous methods of estimation r , and their numerical values will become explicit in the Results section. As there is no explicit solution for λ , the Solver procedure of MicroSoft Excel spreadsheet was used.

3.1.2 Estimation of the carrying capacity (K)

There is no estimation of the carrying capacity of the Chacoan "*Impenetrable*" for the blue-fronted parrot. I assumed that present densities can be considered as a minimum value of K , unless the blue-fronted parrot population is already at its K value, or in a very favorable year. This is so because one would expect that normally the carrying capacity would be somewhere above the present density value, particularly if it is subjected to offtake in the present and has been so in the recent past (the case of the blue-fronted parrot). In the logistic population model here used (see below) the population was expressed in terms of density of fledglings (number of individuals per km²), so the carrying capacity was also expressed in this same units. However to estimate density of fledglings actually the density of nest-trees and successful nests had to be estimated.

Density of nest-trees was available from the "*Impenetrable*", from basic studies carried out at the "*Loro Hablador* Natural Reserve" protected area (Igor Berkunsky, *pers. comm.*), where nests were located by exhaustive search along roads, paths, trails and tracks in the woods. The search was carried out at the beginning of the reproductive period (usually early Spring, i.e., between the middle and the end of September) and early in the morning or in the afternoon; whenever adult blue-fronted pairs of parrots were heard, the searching group would branch off in the direction of the parrots to get as near as possible to them, and wait quietly until the chicks could be heard. Once a nesting tree was identified, it was marked with a numbered plastic tag, and its position (as well as the path followed) registered with a GPS system. The nests were visited every three-four days for five months, and the number of live and dead chicks was counted, their weight registered, and their length were measured. This procedure was carried out until the end of the reproductive period (approximately mid-February). The aural sign of the singing

pairs was estimated in approximately 250 m on each side of the paths (Igor Berkunsky, *pers. comm.*), so the density of nests and of fledglings surviving until time of harvest was calculated assuming a strip 500 m wide. This detection distance of aural vocalizations of the blue-fronted parrot has been confirmed by other researchers (Esteban Fernández-Jurísic, *pers. comm.*).

3.1.2.1 Direct estimation

By direct estimation it is meant the calculation of a mean density (and its standard deviation) by dividing the number of nest-trees by the area where they were found. This is similar to a strip sampling, and the critical aspect here is to determine the exact surface for the calculation of density, that is, the surface of the “strip”.

In the “Loro Hablador” Natural Reserve, Igor Berkunsky (*pers. comm.*) has carried out a study of the survival of the blue-fronted parrot stages from egg to fledgling. This also permitted the determination of the number of nests at different stages in the chicks’ development, and thus allowed the calculation of fledgling densities. The calculation of nest-trees densities was carried out by dividing the number of nest-trees by a surface area equivalent to the sum of the length of all paths used (27,196 m) multiplied by a width of 500 m (250 m on both sides of the path). Esteban Fernández-Jurísic (*pers. comm.*) even considered that sometimes even at 300 m away blue-fronted parrots can be heard, but not in the same area and habitat of the Chaco; thus the width of 500 m was used. In other words, this direct estimation of densities is equivalent to a strip sampling.

Actually the total of 27,196 m of paths used were rarely a straight line (except in some stretches of roads) and, basically many lateral diversions and detours were made. As a result different “lines” or “transects” were actually made, and these diversions were considered as independent paths so that the direct density was calculated as a mean (and standard deviation) of the densities of the individual paths (or actually strips). In total 17 individual paths were available for the 2002-2003 period and 14 for the 2003-2004 period.

3.1.2.2 Estimation by transects

For the purpose of estimating nest-trees density, the displacement along the roads, paths, and trails in search for nest-trees was assimilated to the transect sampling method (Anderson *et al.*, 1976, Burnham and Anderson, 1976, Burnham *et al.*, 1980). Nest-trees positions along these “transects” were analyzed with the distance sampling method (Eberhardt, 1967, Buckland, 1985, Wilson and Anderson, 1985) using the “Transect” software (Burnham *et al.*, 1980), an earlier version of the “Distance” software (Buckland *et al.*, 1993, Thomas *et al.*, 2003) to estimate densities. When the “branching off” from one trail lead to a another path with several nest-trees, then for the purpose of analysis, this “branched off” trail was considered as an independent transect. Possibly these density estimates may be a biased estimation, for the nest-trees detected were not the result of a random search but a search resulting from the existing roads, paths, and trails that facilitated the search; so they actually represent a “subjective sampling” or “convenience

sampling” (Anderson *et al.* 2001). However, as the habitat was not different between the selected paths and the rest of the environment, it was considered that departures from the condition of randomly established transects would represent a small bias of density estimates (probably representing a subestimation of densities), for large tracks of forest remained unsearched.

Additionally, the detection of the nest-trees did not follow the classical searching scheme along a transect, for several researchers and “*baquianos*” traveled along the paths many times in the same season, adding new nest-trees overlooked previously. However, although this seems to approximate more to a census rather than a sampling scheme, there may always be some nest-trees not detected, and at the most we can consider this procedure as a very exhaustive transect sampling, with a highly efficient coverage of the searched area. Nevertheless, the four main assumptions underlying the correct application the transect sampling methods (Burnham *et al.*, 1980) are fulfilled in the case of the blue-fronted parrot nest-tree sightings in the “*Loro Hablador*” Natural Reserve: (1) nest-trees directly on the path are never missed, (2) nest-trees are fixed at the initial sighting position (they are not “movable objects”) and none are counted twice, (3) no measurement nor counting errors occur, and (4) sightings are independent events.

The probability distribution of blue-fronted parrot detection from the transect to the nest-trees using the parrots’ aural signal is not known, but the Transect software provides a fit to several models of the probability of detection with distance from the transect. I preferred to use the Fourier series model of probability of detection. The Fourier series model is a nonparametric method, and nonparametric approaches for the estimation of densities with the transect method are advantageous because they provide good estimates under much broader circumstances than do the simpler parametric models. Under some distributions the Fourier series is more efficient for sample sizes likely to be encountered in practice than the corresponding parametric estimators. In general the efficiency of the Fourier series estimator used in the “Transect” software was developed to cover a broad set of underlying distributions and compares favorably with simple parametric estimators developed only for very specific distributions (Burnham *et al.*, 1980).

Another advantage of the Fourier series model is to cope with one of the most important assumptions underlying the line transect method: that animals are randomly and independently distributed over the population area (Gates *et al.*, 1968). However Quinn (1977, cited in Gates, 1979) investigated the assumption of randomness by substitution of an aggregated population from a randomly distributed population in a computer simulation model of line transects. The spatial structure of the population was modeled with the use of the negative binomial probability distribution to account for different levels of aggregation. The results showed that no bias is induced in the estimates using the Fourier series model, but that the estimates of the theoretical variance become increasingly biased with an increase in aggregation (Gates, 1979).

The application of the Fourier series model with the Transect program (Laake, Burnham and Anderson, 1979) was carried out using the individual paths as independent transects instead of a single path 27,196 m long. However, this procedure has the disadvantage that

the some lines (transects) are relatively short and the number of nest-tress detected is low. For these situations Burnham *et al.*, (1980) suggest the estimation of the mean and the variance of the density using the Jackknife method. This is a procedure based on (1) recombining the original data, (2) calculating pseudo-values of the parrot density for each recombination of the original data, and (3) estimating the mean value and standard error of the nest-trees density from the resulting frequency distribution of pseudovalues. For a more extensive review of theory behind this procedure, see Gray and Schucany (1972) and Miller (1974).

3.1.2.3 Estimation by nearest neighbor and point-to-nearest object method

The same tree-nest data used in the transect analysis was also analyzed by the nearest neighbor and point-to-nearest object methods, a related technique sometimes used by botanists and foresters to estimate tree densities (Hopkins, 1954). These approaches have the advantage that they do not involve modeling the detection function as with transects. These methods also allow testing the type of dispersion of the nest-trees, to verify if they depart from a spatial random distribution (Byth and Ripley, 1980). To carry out these calculations the mapping of the sampled nest-trees with fledglings that had been recorded with the GPS system was used. As in this mapping that allowed a “virtual” sampling the actual area explored is restricted to the path followed by the collectors and/or the researchers, I had to resort to an additional procedure: around each nest-tree with fledglings a circle was drawn with changing the radius of all nest-trees simultaneously, under the conditions established by a especial algorithm (see Appendix IX for a description of the conceptual and programming details). The location of random points to find the nearest neighbor was restricted to these circles. The rationale behind this procedure was that as the collectors and/or researchers follow an aural signal to detect the nest-trees, the criterion of tangent circles works, to some degree, as an estimate of the radius determined by the aural signal. A special program to read the coordinates, draw the circles and apply the statistical tests was written in C++ language (Appendix X).

To determine if the nest-trees have a random or clumped distribution, I used the *ht* Eberhardt index test (Hines and Hines, 1979):

$$ht = \frac{2n(2\sum x^2 + \sum z^2)}{(\sqrt{2}\sum x + \sum z)^2}$$

where x is the distance from a random point and the nearest nest-tree, and z is distance from that nest-tree to the nearest (neighbor) nest-tree, under the restriction that the angle has to be greater than 90 degrees, and n is the number of random points used (this is also called the T-square method; Krebs, 1989). Values of *ht* greater than 0.5 tend to have a clumped spatial distribution, and they can be tested statistically for significance with a table of the F distribution.

The mean density is given by Byth (1982) as:

$$N_t = \frac{n_2}{2 \sum x \sqrt{2 \sum z}}$$

and the standard error by Diggle (1983) as:

$$SE = \sqrt{\frac{8 \left(\left(\frac{\sum z}{n} \right)^2 sx + 2 \left(\frac{\sum x}{n} \right) \left(\frac{\sum z}{n} \right) sxz + \left(\frac{\sum x}{n} \right)^2 sz \right)}{n}}$$

where

$$sx = \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n-1},$$

$$sz = \frac{\sum z^2 - \frac{(\sum z)^2}{n}}{n-1}, \text{ and}$$

$$sxz = \frac{\sum xz - \frac{\sum x \sum z}{n}}{n-1}$$

with x and z as defined before.

3.1.2.4 Estimation by adaptive sampling

Under the preliminary impression that nest-trees may have a clumped dispersion pattern, I selected the adaptive sampling scheme because it is very efficient in situations where organisms follow that kind of spatial distribution (Thompson, 1991). This approach is based in the following concept: to sample more intensively where aggregations have been identified. I used the GPS data of all nest-trees of the “*Loro Hablador*” Natural Reserve, and designed a computer program to carry out a “virtual” adaptive sampling applying the procedure suggested by McCallum (2000):

- (a) I used the various paths of the nest-trees of the “*Loro Hablador*” Natural Reserve as the area to be sampled.
- (b) To demarcate that area, I developed a program that drew several thousand of 250 m-radius circles, with center along the paths (250 m distance on each side of the path was the accepted distance to detect the aural signals of the blue-fronted parrot).
- (c) I divided that demarcated area into a grid of cells (see below), where the cells represent the units subjected to sampling.
- (d) I took a sample of n “primary” units (see below).

- (e) The number of nest-trees in each “primary” unit is counted (as indicated by their GPS position).
- (f) If the nest-tree number in a “primary” unit is above a certain threshold (I used as threshold a value of zero, which usual in this approach), then all adjacent units are sampled.
- (g) In all adjacent units that have a density above the threshold of zero, their adjacent units are sampled in turn.
- (h) The last step is repeated until no more adjacent units have a density above the threshold density.

This procedure leads to what is called a series of networks of clusters, where a cluster is a group of sampling units that satisfied the threshold condition. If we call κ the number of networks, then the probability α_κ that any one of the units in the κ^{th} network will be used in the estimator of the mean is given by:

$$\alpha_\kappa = 1 - \binom{N-x_\kappa}{n} / \binom{N}{n}$$

We define z_κ as an indicator variable that takes the value of one if any unit of the κ^{th} network is included in the initial sample; z_κ takes the value zero otherwise. Then the estimated average density (y) per quadrat is given by:

$$\mu = \frac{1}{N} \sum_{\kappa=1}^K \frac{y_\kappa^* z_\kappa}{\alpha_\kappa}$$

in which y_κ^* is the total of the y values in the κ^{th} network. The conversion from density per quadrat to density per km^2 was carried as a function of the surface area of the “primary” units. The variance is a much more complicated formula and the operative version used in our program can be found in McCallum (2000).

Quadrats (“primary” units) of several sizes (always square quadrats) and various quadrat numbers were used in various viable combinations (some combinations could never comply with certain quadrat size and quadrat number within the demarcated sampling area): the quadrat sizes were 200, 250 and 300 m in their sides, and the quadrat numbers (the n “primary” units) were 20, 25, 30, 35, and 40. That is, 15 different combinations were used. As the adaptive sampling starts with a series of random initial (“primary”) quadrats and the results would vary with different “runs” of this sampling scheme, 20 repetitions of each of the 15 combinations were carried out to have an idea of the degree of dispersion of the nest-tree density estimates. For each repetition the mean density (nest-trees/ km^2), the standard error, and the 95% confidence limits were calculated. As we could anticipate a certain degree of dispersion of the mean density values among repetitions, the criterion of the minimum coefficient of variation was used to select one result among the 20 repetitions.

3.2 The Logistic Population Growth Model

The logistic population growth model is a very simple but biologically sound model. It is “simple” because, compared to the exponential growth model, it only relaxes the assumption of an unlimited environment, by recognizing that there is a carrying capacity of the environment (K). It makes additional assumptions, such as that the rate of population growth (r) decreases linearly as it approaches K . Despite its simplicity the logistic model has been used in a variety of applications that go from wildlife management of species such as elephants (Basson *et al.*, 1991) to the control of pest populations such as insects (Rossi and Fowler, 2004) or pathogenic viruses (Marsh *et al.*, 2000).

The above-mentioned assumptions when applied to the exponential model result is the classical logistic equation:

$$dN/dt = r N [(K-N)/K]$$

where N is the size of the population, and r is the intrinsic rate of natural increase (also called the instantaneous per capita population growth rate or the instantaneous finite rate of increase), and K is the carrying capacity of the environment for blue-fronted parrots.

This model makes several additional implicit assumptions, such as: (a) the environment is evenly distributed, (b) reproduction occurs at discrete intervals with non-overlapping generations, (c) “partial” (parts of) individuals are possible, (d) individuals are ageless and sexless, and equal in their effects on population size, (e) population is only defined at discrete intervals, (f) there is no dispersal, and (g) there are no time lags (population size “detects” approaching the level of K instantaneously). Despite these flagrant simplifications the logistic population growth model has been well fitted to laboratory and wild populations of many species.

A FORTRAN program was developed (Appendix VIII) to simulate the logistic population growth model of a blue-fronted parrot population in a discrete form, assigning to it different values of its two parameters, r and K . The discrete form I used assumes the following expression:

$$N_{t+1} = N_t \frac{R_{\max} K}{K + [N_t (R_{\max} - 1)]}$$

where, as before, N_t is the population size of the blue-fronted parrot in year t , K is the carrying capacity of the “*Impenetrable*” for the blue-fronted parrot, and R_{\max} is the finite population growth rate and is equal to $\exp(r)$ (also denoted as e^r), and where r is the intrinsic rate of natural increase (Brauer and Sánchez, 1975, Basson, Beddington and May, 1991). The model also included the “*fixed escapement*” rule of population management described below.

3.3 The “fixed escapement” rule of population management

The “*fixed escapement*” rule of population management can be expressed in the following verbal manner: if, at a given time t , the population density (D_t) is equal or less than a certain value called the escapement or threshold density (D_U), then the harvest (Z) is fixed to zero; if population density (D_t) is larger than the escapement density (D_U), then the harvest (Z) is set to the density of animals above (or in “excess” of) the escapement density (D_U).

Another convenient way of expressing this rule is in relative terms, that is, using harvest rates (that is the proportion of the population that is harvested and not the absolute density). In this case if D_t is larger than D_U then harvest Z is determined by the relation $(D_t - D_U)/D_t$. Algebraically this can be summarized as:

$$\begin{aligned} \text{if } D_t \leq D_U &\rightarrow Z = 0 \\ \text{if } D_t > D_U &\rightarrow Z = \frac{D_t - D_U}{D_t} \end{aligned}$$

The above formulae outline one of the main properties of this management rule: its feedback nature. If there is an unanticipated event (e.g., an abnormal dry period or a disease’s epidemic) that results in an unusually low blue-fronted parrot population (in relation to the one expected under normal circumstances), then the harvest of fledglings would be much smaller (and eventually nil). This mechanism is a sort of sustainability assurance for, once the rule’s threshold density has been established, there is no danger that the uncertainty that usually characterizes ecological systems may represent an overexploitation risk. Another advantage of this rule is that it also takes into account the effects of a possible poaching, for if this illegal activity takes place, at the following offtake year the population of blue-fronted parrots would be less than expected, and the harvest would then be reduced accordingly (however, no poaching was simulated in the model here presented). There is a well-developed literature showing that the “*fixed escapement*” rule of population management is an excellent rule to incorporate uncertainty (Ludwig and Walters, 1981; Walters, 1981; Walters, 1986; Hilborn and Walters, 1992, Lande *et al.*, 2001).

The threshold value, so critical for this management scheme, was determined by simulation. The criteria for determining the threshold value are crucial and strongly affect the results. Two of the most common criteria are: (a) to maximize the total accumulated harvest for a given time period, or (b) to maximize the total accumulated discounted profits for a given time period. I preferred the former, to avoid getting involved (at least in this preliminary model) into the economics of the blue-fronted parrot management. For these simulations a 30 years time horizon was selected to run the program with the logistic population growth and its “*fixed escapement*” rule, with the number of harvested fledglings being recorded each simulated year and accumulated for 30 years. The order of the operations was to first let the population “grow” and then “apply” the harvest. The basic logistic model without harvest becomes now:

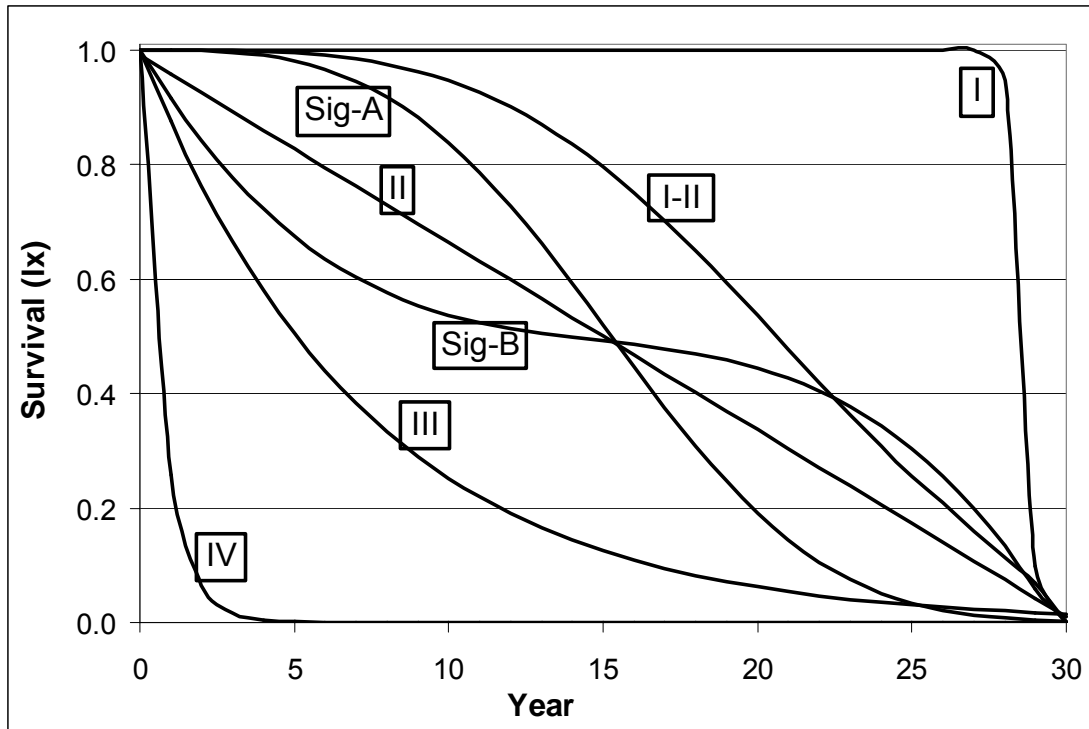


Fig. 6. Hypothetical survival curves, generated with Weibull and polynomial functions, to estimate the intrinsic rate of natural increase (r) of the blue-fronted parrot (*A. aestiva*). See text for development, interpretation, and use.

Fig. 7 shows the polynomial fit of the Binkley and Miller (1980) method (MB-80) with its respective mortality rate curve.

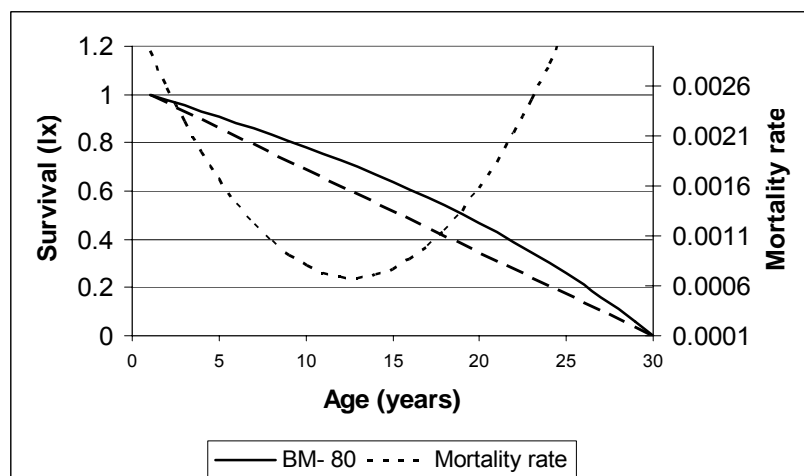


Fig. 7. Results of the Binkley and Miller (1980) method to calculate the BM-80 survivorship curve (l_x) after calculating the mortality rates (d_x). The diagonal dashed straight line was drawn only for reference purpose, to show that the BM-80 l_x curve lies slightly above survivorship curve type II.

The average longevity calculated from the survivorship curves of Figs. 6 and 7 are shown in Table 2.

Table 2. Average longevity calculated from the survivorship curves of Figs. 5 and 6. For shaded cells and bold numbers see text.

Curve type	Average longevity (years)	Curve type	Average longevity (years)
I	28.95	I-II	22.34
II	14.51	Sig-A	14.83
III	6.64	Sig-B	14.21
IV	0.35	MB-80	17.38

An average longevity of 29 years (from curve type I) was considered excessive (being so close to the average maximum longevity, with which it should not be confused), and the average longevity of 0.3 years (from curve type IV) was considered senseless for the blue-fronted parrot; similarly curve type III resulted in an average longevity of 6.6 years, and although not an unreasonable figure, it is considered to be too low for the blue-fronted parrot (Ricardo Banchs, Igor Berkunsky, and Eugeni Fernández Nogales, *pers. comm.*). Additionally, the average longevity for the blue-fronted parrot estimated by regression from weight and reach (see Appendix IV) oscillates around 20-21 years, so it was considered that survivorship curves that yielded *average* longevity between 14 and 22 years would be the most plausible ones (i.e., curves types II, I-II, Sig-A, Sig-B, and MB-80). Survivorship curve types I, III, and IV were anyway used for the calculation of the intrinsic rate of natural increase (r) only for the sake of completeness, but only the survivorship curve types identified with a shaded and bold font are considered to be the ones that may provide a reasonable estimate the intrinsic rate of natural increase (r). Of these, the ones with the figures in bold were considered as the most credible ones.

The hypothetical values of the survivorship curves of Fig. 5 represent only the adult survival (from absolute year one, onwards), and had to be scaled so that the initial value would incorporate fledgling mortality. Development mortality up to the fledgling stage was based on unpublished data from the “*Loro Hablador*” Natural Reserve (Igor Berkunsky, *pers. comm.*) and was used to calculate the average survival from the egg stage to the fledgling offtake age (last row of Table 3).

The average survival estimated from years 2002 and 2003 from egg to offtake age (0.586) was used as the initial value of the survivorship curves (Figs. 5 and 6), and these curves were used to estimate the intrinsic rate of natural increase (r) in conjunction with the maternity function (age-specific reproduction schedule) using the Lotka equation.

The age of first reproduction (α) was estimated from the data of Sæther and Bakke (2000) using a linear multiple regression. The resulting best fit equation was given by: $-3.9905 + 0.2595 \text{ Fec} + 0.9093 \text{ Sj} + 8.5534 \text{ Sa}$, where Fec, Sj, and Sa stand for fecundity, juvenile survival and adult survival, respectively (see Methods). The coefficient of multiple correlation was 0.7267, and the regression was statistically significant, with $F(3,35)=13.057$ $p<.00001$ ($N=39$). Applying to this equation the best data available for *A. aestiva* ($F= 0.986$ female offspring per female per reproductive season (see last paragraph

of this section); $S_j = 0.58$ (See Table 3), and $S_a = 0.95$ and 0.99 , see Section 4.1.1.4), I obtained $\alpha = 4.9$ years using $S_a = 0.95$ and $\alpha = 5.2$ years using $S_a = 0.99$. As there were no first hand field estimates of α for the blue-fronted parrot I decided to use a fairly wide range of ages of first reproduction (α values of 1, 2, 4 and 6 years) to estimate r .

Table 3. *A. aestiva* survival estimates from the egg to fledgling at the “Loro Hablador” Natural Reserve (Igor Berkunsky, *pers. comm.*). Incubation survival was not measured in 2002, and hatching and brooding survival in 2002 is not very reliable (based upon only 5 nests). The standard deviation of incubation survival in year 2003 is zero because minimum and maximum values were one. Averages for years 2002 and 2003 are only indicative, to be used as the initial survival value of the adult stage in the survivorship curves (Figs. 5 and 6), and not to assign a solid estimate to the different survival values.

	Year						Mean 2002-2003
	2002			2003			
	Mean	Std. Dev.	n	Mean	Std. Dev.	n	
Incubation	-	-	-	1.00	0.00	14	-
Hatching	0.63	0.04	5	0.70	0.20	17	0.67
Brooding	0.93	0.15	5	0.93	0.14	18	0.93
Age at offtake	0.99	0.07	24	0.91	0.21	23	0.95
Fledglings	0.99	0.07	23	0.90	0.24	13	0.95
From egg to fledgling	0.57			0.53			0.55
From egg to offtake age	0.58			0.59			0.586

4.1.1.1 The Lotka equation

Table 4 shows the results of r estimated for different “skipping rates” and different ages of first reproduction (α) as calculated with the Lotka equation. The average m_x (number of eggs laid per female per year, or clutch size) value used was 3.944 (std. dev. 0.938, $n = 18$; Igor Berkunsky, *pers. comm.*). This clutch size value was corrected by two factors in order to be considered the maternity function: (i) the proportion of successful nests (the value of 0.5, corresponding to 2003, was used; Igor Berkunsky, *pers. comm.*, similar to the values 0.50-0.62 obtained by Banchs *et al.* (2000)), and (ii) the proportion of females among offspring (also 0.5, assuming a 1:1 sex ratio). See the final paragraph of Section 3.1.1.1. and Appendix VII for a justification of the 1:1 sex ratio used.

Thus the final m_x used for the estimation of r was 0.986 ($= 3.944 \times 0.5 \times 0.5$; i.e., near one female egg per pair per year). Although the information from aviaries indicates that in the first and the last 4-5 reproductive years fecundity is somewhat lower than average, I used a constant 0.986 value for m_x . The application of the “skipping rates” was straight forward, applying the value 0.986 of the m_x function every year for a “skipping rate” of zero, every two years for a “skipping rate” of one, and every three years for a “skipping rate” of two.

Table 4. Intrinsic rate of natural increase (r) estimated from the Lotka equation, and based upon various survivorship curve types, for different ages of first reproduction (α) and “skipping rates”. For the age-specific reproduction schedule see description in text. The shaded cells of the table represent the most plausible combinations of “skipping rate”, age of first reproduction (α), and survivorship curves. The values in bold result from the selection of the most “credible” combination of parameter values and survivorship curves.

Survivorship curve type I				Survivorship curve type II			
Skipping rate				Skipping rate			
α	0	1	2	α	0	1	2
1	0.570	0.335	0.239	1	0.520	0.199	0.192
2	0.389	0.209	0.160	2	0.345	0.206	0.142
4	0.262	0.181	0.141	4	0.218	0.135	0.093
6	0.205	0.145	0.188	6	0.159	0.097	0.064
Survivorship curve type III				Survivorship curve type IV			
Skipping rate				Skipping rate			
α	0	1	2	α	0	1	2
1	0.398	0.174	0.083	1	-1.074	-1.212	-1.266
2	0.234	0.100	0.039	2	-1.135	-1.235	-1.278
4	0.115	0.035	-0.005	4	-1.196	-1.264	-1.297
6	0.060	0.001	-0.031	6	-1.229	-1.283	-1.312
Survivorship curve type I-II				Survivorship curve type SIG-A			
Skipping rate				Skipping rate			
α	0	1	2	α	0	1	2
1	0.570	0.335	0.237	1	0.567	0.326	0.223
2	0.389	0.250	0.185	2	0.381	0.237	0.168
4	0.260	0.176	0.134	4	0.245	0.156	0.110
6	0.200	0.138	0.105	6	0.177	0.110	0.074
Survivorship curve type SIG-B				Survivorship curve type BM-80			
Skipping rate				Skipping rate			
α	0	1	2	α	0	1	2
1	0.467	0.248	0.162	1	0.553	0.315	0.216
2	0.304	0.176	0.120	2	0.367	0.228	0.163
4	0.192	0.117	0.079	4	0.237	0.153	0.111
6	0.144	0.087	0.057	6	0.176	0.114	0.081

Although the values of r in Table 4 fall roughly within the range 0.1-0.4, if we assume that the most reasonable values for the age of first reproduction (α) stand between 2 and 6 years, and that of the “skipping rates” to 1 and 2, and restricting the combinations to the survivorship curves type II, III, I-II, Sig-A, Sig-B, and BM-80 (that is excluding type I and IV), then the most plausible values of the intrinsic rate of natural increase (r) would fall roughly (rounding to two decimal points) within the range 0.1-0.25 (shaded values in Table 4). However, if we would like to get a finer approximation I would claim that a “skipping rate” of 2 and a of 4 would probably be the most credible values, which would results in r values between 0.12 and 0.18 (figures in bold in Table 4).

4.1.1.2 Estimation of r using Cole's formula

The results of the estimation of the intrinsic rate of natural increase (r) using Cole's equation are shown graphically in Fig. 8, and as a table in Table 5. A clutch size (female eggs per female per year) of one results from assuming 2 female eggs per pair with a "skipping rate" of one. Increasing the age of last reproduction (ω) has very little effect on the estimation of r , after a certain value of ω has been reached. This is usual in some species, like the blue-fronted parrot, with an early age of first reproduction relative to its longevity.

Table 5. Intrinsic rate of natural increase (r) estimated from Cole's equation (Cole, 1954). The shaded cells of the table represent the value of r from the most plausible combinations of parameters α , ω , and b . Dashes are "impossible" combinations, and zeroes imply r values below 0.001. The shaded cells are the results from the selection of the most "plausible" parameter values. The values in bold are the results from the selection of the most "credible" parameter values.

Age of first reproduction (α) (in years)	Clutch size (female eggs per pair per year)	Age of last reproduction (ω) (in years)					
		5	10	15	20	25	30
1	2	0.61	0.69	0.69	0.69	0.69	0.69
	1	0.21	0.39	0.40	0.41	0.41	0.41
2	2	0.00	0.46	0.48	0.48	0.48	0.48
	1	0.00	0.27	0.30	0.31	0.31	0.31
4	2	-	0.00	0.30	0.32	0.32	0.32
	1	-	0.00	0.19	0.22	0.22	0.23
6	2	-	0.00	0.16	0.23	0.25	0.25
	1	-	0.00	0.06	0.15	0.17	0.18

The effects of the age of first reproduction (α) are more important, as shown in Fig. 8 and in Table 5. Increasing the age of first reproduction (α) (i.e., delaying reproduction) has a dramatic effect in reducing the estimated value of r . Note how the curves with r values, as they are expressed in the same scale, become lower as the age of first reproduction (α) is delayed.

4.1.1.3 Estimation of r using generation time and replacement rate

Table 6 shows the generation time (T_c) for all survivorship curve types and four ages of first reproduction (α) (we implicitly assumed a "skipping rate" of zero), and Table 7 shows the replacement rate (R_o) for all survivorship curve types and four ages of first reproduction (α).

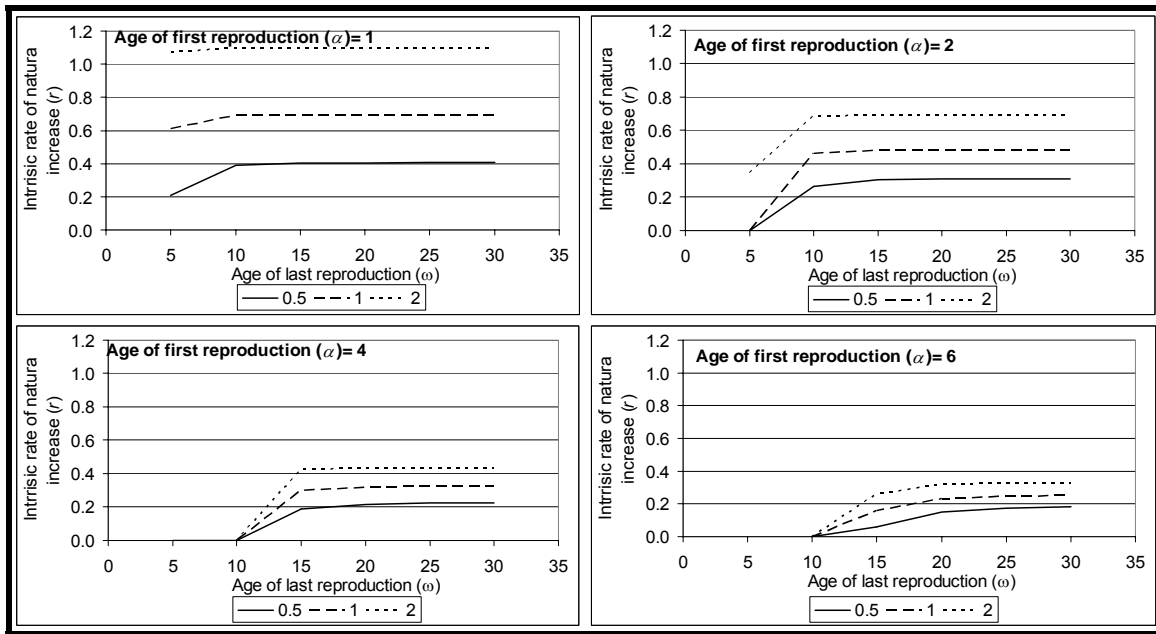


Fig. 8. Estimation of the intrinsic rate of natural increase (r) using Cole's equation for different combinations of ages of first (α) and last (ω) reproduction.

Table 6. Generation time (T_c) (years) for different survivorship curve types and ages of first reproduction (α). Shaded cells are the most plausible combinations.

Survivorship Curve type	Age of first reproduction (α) (years)			
	1	2	4	6
I	14.98	15.48	16.48	17.48
II	10.42	11.09	12.42	13.76
III	7.16	8.10	9.95	11.78
IV	1.34	2.34	4.34	6.34
I-II	12.19	12.72	13.77	14.84
SIG-Sup	8.83	9.40	10.56	11.78
SIG-Inf	11.89	12.64	14.04	15.34
BM-80	11.17	11.79	13.03	14.28

Table 7. Replacement rate (R_o) for different survivorship curve types and ages of first reproduction (α). Shaded cells are the most plausible combinations.

Survivorship Curve type	Age of first reproduction (α) (years)			
	1	2	4	6
I	16.74	16.16	15.00	13.85
II	8.38	7.83	6.78	5.80
III	3.84	3.34	2.51	1.88
IV	0.20	0.05	0.00	0.00
I-II	12.91	12.34	11.18	10.02
SIG-Sup	8.57	7.99	6.84	5.70
SIG-Inf	8.22	7.69	6.75	5.94
BM-80	10.05	11.79	13.03	14.28

Table 8 shows the intrinsic rate of natural increase (r) for the combinations of all survivorship curve types and the four ages of first reproduction (α).

Table 8. Intrinsic rate of natural increase (r) for different survivorship curve types and ages of first reproduction (α) using generation time and replacement rate. Shaded cells are the most plausible combinations, and in bold the most credible ones.

Survivorship curve type	Age of first reproduction (α) (years)			
	1	2	4	6
I	0.19	0.18	0.16	0.15
II	0.20	0.19	0.15	0.13
III	0.19	0.15	0.09	0.05
IV	-1.20	-1.27	-1.32	-1.33
I-II	0.21	0.20	0.18	0.16
SIG-Sup	0.24	0.22	0.18	0.15
SIG-Inf	0.18	0.16	0.14	0.12
BM-80	0.21	0.21	0.20	0.19

4.1.1.4 Two ages matrix approach

For the application of this method the following values of the demographic parameters of the blue-fronted parrot were used: b (female progeny that fledged per reproductive female)= 0.986, c_o (proportion of yearlings attempting breeding)= 0, c_a (proportion of adults (2 years old or older) attempting breeding)= from 0.5 to 1, s_o (juvenile survival, i.e., during the first year of life)= 0.5, and s_a (overall adult annual survival)= 0.99. The value of b corresponds to the clutch size (3.944 eggs per pair), divided by two to restrict b to the females of the population (i.e., a 1:1 offspring sex ratio is assumed), and divided by two again because the proportion of successful nests is 0.5. The parameter c_o was set at zero, for no fledglings reproduce. The parameter c_a (proportion of adults that breed) was assigned six values between 0.5 and 1 with steps of 0.1 (0.5, 0.6, 0.7, 0.8, 0.9, and 1). Parameter s_o (annual survival of yearlings) was set to 0.58 (see Table 3), and parameter s_a was estimated in 0.99. The latter value, the overall annual adult population survival (s_a), was estimated by a two step procedure: (a) the BM-80 l_x curve was used with an m_x curve, and the Stable Age Distribution of a population with 30 individuals of up to 30 years of age was calculated, (b) the mortality rate curve BM-80 (see Fig. 6) was used to estimate the overall annual population survival as a weighed average using the Stable Age Distribution (as a proportion) as the weight for each year class). It was seen that the age of first reproduction (α) had little influence on the overall annual population survival (0.9968 for $\alpha= 1$, 0.9979 for $\alpha= 2$, 0.9984 for $\alpha= 3$, 0.9987 for $\alpha= 4$, and 0.9989 for $\alpha= 5$), so the value 0.99 was used.

Table 9 shows the results of the intrinsic rate of natural increase (r) estimated for five values of c_a . High values of c_a (0.8, 0.9, and 1) are probably quite unrealistic, and the values of r for lower c_a values (the shaded rows in Table 8) are deemed as the most plausible ones.

Table 9. Finite rate of growth (λ_1) and the intrinsic rate of natural increase (r) estimated by the two ages matrix method, for different values of c_a (proportion of adults that breed in a given year).

c_a	λ_1	r
0.5	1.403	0.339
0.6	1.464	0.381
0.7	1.521	0.420
0.8	1.576	0.455
0.9	1.627	0.487
1.0	1.677	0.517

The stable age distribution is shown in Table 10, also for the various values of c_a used to estimate r , showing, as expected, higher proportions of yearlings the higher the growth rate. For the most plausible values of c_a , the proportion of adults in the population is a rounded value of approximately 0.7.

Table 10. Stable age distribution of the two ages matrix model, for various values of c_a

c_a	Yearlings	Adults
0.5	0.30	0.70
0.6	0.33	0.67
0.7	0.35	0.65
0.8	0.38	0.62
0.9	0.40	0.60
1.0	0.41	0.59

Table 11 shows the various effects of the demographic parameters on the estimation of the value of the finite rate of growth (λ_1) and hence on the intrinsic rate of natural increase (r). The upper part of Table 11 shows the (absolute) sensitivities of λ_1 , while the lower part of the table shows the elasticities, which were calculated (for each demographic parameter) as their respective sensitivities multiplied by the value of the demographic parameter and divided by the finite rate of growth (λ_1). It becomes clear that both sensitivities and elasticities point towards a dominant role of the adult survival rate in determining the value of the intrinsic rate of natural increase (r). This had already been suggested for parrot species by Beissinger (2001).

Table 11. Sensitivities and elasticities of λ_1 (and r) to the various demographic parameters of the two ages matrix model. Parameter b is the female progeny that fledged per reproductive female, c_o is the proportion of yearlings attempting breeding, c_a is the proportion of adults (2 years or older) attempting breeding, s_o is the juvenile survival, i.e., during the first year of life, and s_a is the overall adult annual survival. Shaded cells represent the most plausible values of c_a . Cells in bold represent the results for the most credible of the plausible values of c_a .

	Sensitivities					
	c_a (proportion of adults that breed)					
	0.5	0.6	0.7	0.8	0.9	1
b	0.157	0.176	0.194	0.210	0.225	0.240
c_o	0.263	0.283	0.299	0.313	0.325	0.335
c_a	0.618	0.579	0.546	0.518	0.494	0.473
s_o	0.533	0.599	0.659	0.714	0.766	0.816
s_a	1.098	1.121	1.143	1.165	1.187	1.208

	Elasticities					
	c_a (proportion of adults that breed)					
	0.5	0.6	0.7	0.8	0.9	1
b	0.230	0.247	0.261	0.274	0.284	0.293
c_o	0.000	0.000	0.000	0.000	0.000	0.000
c_a	0.230	0.247	0.261	0.274	0.284	0.293
s_o	0.230	0.247	0.261	0.274	0.284	0.293
s_a	0.770	0.753	0.739	0.726	0.716	0.707

4.1.1.5 Homeotherm meta-analysis

The application of body mass vs. r regression method ($r = aW^n$) is straight forward using the regression coefficients provided by Fenchel's (1974) ($a = -1.4$, $n = -0.275$). Assigning an average weight of 0.5 kg to an adult blue-fronted parrot, the value of the intrinsic rate of natural increase (r) results in approximately 0.3 (exactly 0.298).

4.1.1.6 Lande's equation

The application of the Solver procedure of the MicroSoft Excel spreadsheet to obtain a solution for λ_1 involved providing the values of the other four demographic parameters of Lande's equation. The annual adult female survival (s) was estimated from two survivorship curves: one that assumed a constant number of survivors per year (type II) and one that assumed a constant rate of surviving per year (type III). In both cases the calculations were also carried out with the Solver procedure of the MicroSoft Excel spreadsheet to obtain an overall annual survival rate such that the number of individuals surviving at year 30 would be a non-negative minimum. Curve type III was disregarded because it produced an overall annual survivorship rate of 0.570, which implied an average life span of the blue-fronted parrot of only 3 years. The Solver procedure applied to curve type II resulted in an overall annual survivorship rate of 0.966, which implied an

Table 12. Intrinsic rate of natural increase (r) estimated from Lande's equation, and based upon various survivorship curve types, for different ages of first reproduction (α) and various adult overall survival. The shaded cells of the table represent the most plausible combinations of adult overall survival, age of first reproduction (α), and survivorship curve. The values in bold result from the selection of the most "credible" pair of parameter values and of survivorship curve.

Survivorship curve type I				Survivorship curve type II			
α	Adult Overall Survival			α	Adult Overall Survival		
	0.95	0.97	0.99		0.95	0.97	0.99
1	0.554	0.566	0.577	1	0.550	0.562	0.573
2	0.394	0.404	0.414	2	0.383	0.393	0.403
4	0.266	0.275	0.283	4	0.249	0.259	0.267
6	0.207	0.214	0.222	6	0.187	0.196	0.204
Survivorship curve type III				Survivorship curve type IV			
α	Adult Overall Survival			α	Adult Overall Survival		
	0.95	0.97	0.99		0.95	0.97	0.99
1	0.496	0.508	0.520	1	0.302	0.316	0.331
2	0.326	0.337	0.348	2	0.122	0.137	0.152
4	0.190	0.200	0.210	4	0.001	0.019	0.035
6	0.126	0.136	0.146	6	-0.035	-0.015	0.003
Survivorship curve type I-II				Survivorship curve type SIG-A			
α	Adult Overall Survival			α	Adult Overall Survival		
	0.95	0.97	0.99		0.95	0.97	0.99
1	0.554	0.566	0.577	1	0.554	0.566	0.577
2	0.394	0.404	0.414	2	0.394	0.403	0.413
4	0.266	0.275	0.283	4	0.265	0.273	0.282
6	0.206	0.214	0.222	6	0.204	0.211	0.219
Survivorship curve type SIG-B				Survivorship curve type BM-80			
α	Adult Overall Survival			α	Adult Overall Survival		
	0.95	0.97	0.99		0.95	0.97	0.99
1	0.515	0.527	0.539	1	0.554	0.566	0.577
2	0.350	0.360	0.371	2	0.390	0.400	0.409
4	0.219	0.229	0.238	4	0.259	0.267	0.276
6	0.160	0.170	0.178	6	0.197	0.205	0.213

average life span of the blue-fronted parrot of 16 years, that conforms satisfactorily with the results of Table 2. Nevertheless, I used three values for s : the overall annual survivorship rate of 0.97, and one lower (0.95) and one above (0.99) this value (see Table 12). As before, the female's age at first breeding (α) was assigned four values: 1, 2, 4 and 6 years. The value of l_x (the probability of a fledgling surviving to α) was assigned the corresponding values from each survivorship curve type for each α . Finally, in the "Loro Hablador" Natural Reserve study for 2003 (the most reliable one from the point of view of fledgling's survival) the blue-fronted parrot has a productivity rate of 1.35 (average number of fledglings that are produced alive per pair per year; Igor Berkunsky, *pers. comm.*) so the demographic parameter b (the productivity per female per year) was assumed to be 1.35.

Table 12 shows that, for the most plausible survivorship curves, the values of the intrinsic rate of natural increase (r) estimated from the Lande equation, and based upon various ages of first reproduction (α) and various adult overall survival (s) result in values between 0.23 and 0.28.

4.1.2 Estimation of the carrying capacity (K)

4.1.2.1 Direct estimation of fledgling density

Table 13 shows the results of nest-trees density using the direct estimation method, and Table 14 their use to produce an estimate of the total fledgling density (per km²) by multiplying the nest-trees density by the proportion of successful nests, the proportion of fledglings surviving until offtake time, and the clutch size (see Table 3).

Table 13. Density of active nests at the beginning of the reproductive period and at offtake time in the “Loro Hablador” Natural Reserve. Results for calculations from the direct estimation method. Data from the “*Loro Hablador*” Natural Reserve (Igor Berkunsky (*pers. comm.*)).

	Nest density (per km ²)			
	2002		2003	
	Active nests	With fledglings of offtake age	Active nests	With fledglings of offtake age
Mean	3.6	3.1	2.6	1.7
Std. dev.	2.3	2.4	2.3	2.3
n (transects)	18	18	14	14
n (nests)	38	32	29	22
Lower 95% CL	0.30	-1.80	-0.66	-1.72
Upper 95% CL	6.82	7.93	5.79	5.10

The calculations for Table 14 were carried out with the mean values of all parameters, as well as with the lower and upper 95% confidence intervals, to produce a 95% confidence interval to the total fledgling density.

It can be observed that the 95% confidence intervals are relatively broad (from almost zero to about 14 fledglings/km²). It is interesting that the same amplitude of values was obtained when the 95% confidence intervals were used to calculate a final band for the fledgling density using the RiskCalc software. This software is based on the fuzzy set algebra rules, where the variables that participate in the calculations are given as a range of values instead of single numerical values, and the results are also given as a range of values. Providing the 95% confidence intervals of each variable (i) to (iv) as defined in the header of Table 14, the number of fledglings/km² produced by the RiskCalc software were: [0.678-11.464] and [-1.57-13.803], for 2002 and 2003, respectively.

Table 14. Average density of fledglings (per km²) in the “Loro Hablador” Natural Reserve calculated as the product of its four components: (i) average density by direct estimation of active nests at the beginning of the reproductive period, (ii) average proportion of nests that are successful until offtake time, (iii) average fledgling survival until offtake, and (iv) average clutch size (Igor Berkunsky, *pers. comm.*). Density of active nests correspond to the direct estimation given in Table 12.

Year	Statistic	Density of active nests	Prop. of successful nests until offtake	Prop. of fledgling surviving until offtake	Clutch size	Total fledgling density
2002	Mean	3.56	0.395	0.58	3.944*	3.17
	Lower 95% CL	0.300	0.296	0.390	3.478	0.120
	Upper 95% CL	6.818	0.468	0.811	4.410	11.413
2003	Mean	2.56	0.671	0.59	3.944	4.00
	Lower 95% CL	-0.660	0.606	0.426	3.478	-0.592
	Upper 95% CL	5.789	0.676	0.795	4.410	13.720

*The value of clutch size was not estimated for 2002; the average 2003 clutch size was used (Mean= 3.944, Std. Dev.= 0.938, N= 18).

Although not of direct concern for this model, it was of interest to have an idea of the *total* (fledglings, juveniles and adults) population density of the blue-fronted parrot in the area of study of the Dry Chaco. For that purpose, based upon the nest density estimated from the direct method, we can proceed as follows: the blue-fronted parrot is not a co-operative breeder, so there is only one pair (female and male) per nest. As not all the adult population reproduces each year (c_a , see Section 4.1.1.4), and as the population is composed by a mixture of adults and juveniles (the latter being those individuals of two and onward years of age that still have not performed their first reproduction, and its proportion denoted c_j), we can calculate the total population density, based upon c_a and c_j .

Table 15. Total density of the blue-fronted parrot population (parrots/km²) calculated from density of active nests estimated by the direct method, assuming various proportions of adults reproducing (c_a) and various proportions of juveniles (c_j) in the population.

Year	Proportion juveniles (c_j)	Proportion of adults reproducing (c_a)			
		0.5	0.6	0.7	0.8
2002	0.2	17.08	14.23	12.20	10.68
	0.25	17.79	14.83	12.71	11.12
	0.3	18.51	15.42	13.22	11.57
	0.35	19.22	16.01	13.73	12.01
2003	0.2	12.31	10.26	8.79	7.69
	0.25	12.82	10.68	9.16	8.01
	0.3	13.33	11.11	9.52	8.33
	0.35	13.85	11.54	9.89	8.65
Average 2002/2003	0.2	14.70	12.25	10.50	9.18
	0.25	15.31	12.76	10.93	9.57
	0.3	15.92	13.27	11.37	9.95
	0.35	16.53	13.78	11.81	10.33

However, the demographic parameters c_a and c_j are not well known for the blue-fronted parrot, so I have calculated the total blue-fronted parrot population density for a combination of possible values of c_a , and c_j , and carried out the estimation with the nest densities of years 2002, 2003 and the average between them. These results are shown in Table 15. Depending upon the values of the demographic parameters c_a and c_j , the density of the *total* population of blue-fronted parrot in the "Impenetrable" is between 8 and 19 parrots/km².

4.1.2.2 Estimation of fledgling density by the transect method

The results from the Transect analysis are shown in Table 16. The data used for this analysis was the right angle distances of the nest-trees to the path from which the nest-tree was detected. Song signal's detection is very sensitive to the individual collector and investigator. Despite Mónica Martella's (*pers. comm.*) reservation, a 250 m maximum detection of the aural signal was used, which is the maximum distance at which apparently the blue-fronted parrot can be heard (Igor Berkunsky, *pers. comm.*; Esteban Fernández-Jurísic, *pers. comm.*).

The analysis with the transect method was carried out for years 2002 and 2003 independently. In year 2002 there were a total of 17 transects (summing to 25.28 km) and 38 active nests were found, in year 2003 there were a total of 14 transects (summing to 19.84 km) and 29 active nests were found.

Table 16. Density of nest-trees (nests/km²) estimated by the transect method, using the Fourier series detection distribution, and the Jackknife procedure to estimate variances. Calculations were performed using a 250 m maximum detection of the aural signal (i.e., assuming a strip of 500 m), for years 2002 and 2003. For the number of nests, and the number and length of transects, see text.

Year	Density	Variance	Lower 95% Conf Lim	Upper 95% Conf Lim
2002	14.20	10.60	7.30	21.10
2003	10.99	8.80	4.70	17.28

The conversion of nest densities to fledgling densities implies performing the same operation as given in Table 14, but replacing the value of the density of active nests provided by Igor Berkunsky (*pers. comm.*) by the Transect analyses results of Table 16. The resulting values are shown in Table 17.

Table 17. Average densities of fledglings (per km²), and their lower and upper 95% confidence limits (CL) in the "Loro Hablador" Natural Reserve for years 2002 and 2003, estimated by the transect method, using the Fourier series detection distribution, and the Jackknife procedure to estimate variances. The fledgling density was calculated from nest-tree density as in Table 13.

Year	Average density	Lower 95% CL	Upper 95% CL
2002	12.67	6.51	18.82
2003	17.13	7.33	26.94

4.1.2.3 Estimation of fledgling density by nearest neighbor and point-to-nearest object method

The results from the nearest neighbors analyses are shown in Table 18. For the data available of the “*Loro Hablador*” Natural Reserve the Byth & Ripley method is not very reliable because only six sampling squares were possible for year 2002 and only five sampling squares were possible for year 2003; this problem is evidenced in the very high standard errors of the estimated mean. On the other hand for year 2002 the T-square method allowed the use of up to 30 random points, and resulted an average nest-tree density of 5.05 nest-trees/km² for 2002, and of 2.85 nest-trees/km² for 2003. The lower and upper 95% confidence limits were 3.72 and 7.86 nest-trees/km² for 2002, and 2.05 and 4.67 nest-trees/km², for 2003. Curiously, for the Byth & Ripley sampling, after checking the Hopkins’ Test Index with an *F* table, the results showed a uniform dispersion pattern of nest-trees, probably due to the small number of samples. However, the Hines and Hines *h_T* Test Index for the T-square sampling resulted in a statistically significant aggregated pattern of dispersion for years 2002 and 2003.

Table 18. Mean density of nest-trees (nests/km²) and their 95% confidence limits in the “*Loro Hablador*” Natural Reserve estimated by Byth and Ripley nearest point-to-tree and the T-square nearest neighbor Distance methods.

Year	Method	Density	Standard Error	Lower 95% Conf Lim	Upper 95% Conf Lim	Randomness test
2002	Byth & Ripley***	2.15	0.827	1.23	3.07	0.947*
	T-square****	5.05	0.034	3.720	7.86	1.93**
2003	Byth & Ripley*****	2.85	0.666	1.26	4.41	0.781*
	T-square****	2.85	0.065	2.05	4.67	1.79**

* Hopkins’ Test Index. ** Hines and Hines *h_T* Test Index. *** A maximum of 6 squares could fit in the area delimited by a radius of 250 m from the transects, with an average of 5 trees per square. **** 30 random points could fit in the area delimited by a radius of 250 m from the transects. ***** A maximum of 5 squares could fit in the area delimited by a radius of 250m from the transects, with an average of 4 trees per square.

As with the previous two density estimation methods, the conversion of nest densities to fledgling densities implies performing the same operation as given in Table 14, but replacing the value of the direct density of active nests provided by Igor Berkunsky (*pers. comm.*) by the nearest neighbor analyses results. The results are given in Tale 19.

For year 2002 the T-square (N= 26 random points) resulted in an average density of 5.05 fledglings/km², and of 2.85 fledglings/km² for 2003. The lower and upper 95% confidence limits were 3.72 and 7.86 fledglings/km² for 2002, and 2.05 and 4.67 fledglings/km², for 2003.

Table 19. Density of fledglings (per km²) and their 95% confidence limits calculated from density of active nests estimated by estimated by the Byth and Ripley nearest point-to-tree and the T-square nearest neighbor Distance methods. The other columns and their source as in Table 13. Data from the “*Loro Hablador*” Natural Reserve (Igor Berkunsky, pers. comm.).

Year	Statistic	Density of active nests	Prop. of successful nests until offtake	Prop. of fledgling surviving until offtake	Clutch size	Total fledgling density
Byth & Ripley nearest point-to-tree analysis						
2002	Mean	2.14	0.40	0.58	3.944*	1.94
	Lower 95% CL	1.23	0.30	0.39	3.48	0.49
	Upper 95% CL	3.07	0.47	0.81	4.41	2.05
2003	Mean	2.84	0.67	0.59	3.94	4.44
	Lower 95% CL	1.26	0.61	0.43	3.48	1.13
	Upper 95% CL	4.41	0.68	0.80	4.41	10.44
T-square nearest neighbor analysis						
2002	Mean	5.05	0.40	0.58	3.944*	4.56
	Lower 95% CL	3.72	0.30	0.39	3.48	1.49
	Upper 95% CL	7.86	0.47	0.81	4.41	6.23
2003	Mean	2.85	0.67	0.59	3.944	4.45
	Lower 95% CL	2.05	0.61	0.43	3.48	1.84
	Upper 95% CL	4.67	0.68	0.80	4.41	11.07

*The value of clutch size was not estimated for 2002; the average 2003 clutch size was used (Mean= 3.944, Std. Dev.= 0.938, N= 18).

4.1.2.4 Estimation by adaptive sampling

The results from the adaptive sampling scheme were made up of 15 combinations in terms of number and size of quadrats. Table 20 shows the results of the average nest-tree density and the 95% confidence limits only for the values of the 20 repetitions of each combination that complied with the condition of having the minimum coefficient of variation. From these I selected the nest-trees densities that also represented a combination of Side and N that also had minimum CV (shaded rows).

Table 20. Average nest-tree density (nest-tree/km²) and 95% confidence based upon the adaptive sampling approach. Only one selected value of the 20 repetitions that complied with the condition of having the minimum coefficient of variation (CV) is shown, and not the 15 combinations of number (N) and size (Side) of quadrats available. The shaded rows represent the selected combination of Side and N that also had minimum CV.

Year	Side (m)	N	CV(%)	Average	Lower 95%CL	Upper 95%CL
2002	200	40	0.51%	4.43	4.28	4.57
	250	20	0.97%	6.61	6.38	6.84
	300	20	0.75%	5.56	5.31	5.82
2003	200	35	1.28%	6.53	6.37	6.70
	250	35	1.33%	6.63	6.46	6.81
	300	40	1.28%	4.22	4.11	4.32

Table 21 uses the information of Table 20 to calculate the density of fledglings (per km²) and their 95% confidence limits calculated, based only on the selected shaded rows (minimum coefficient of variation). The other columns and their source are as in Table 13.

Table 21. Density of fledglings (per km²) and their 95% confidence limits calculated from density of active nests estimated by the adaptive sampling method. Row one for year 2002 and row three for year 2003 of Table 20 were used (minimum coefficient of variation). The other columns and their source as in Table 13. Data from the “*Loro Hablador*” Natural Reserve (Igor Berkunsky, *pers. comm.*).

Year	Statistic	Density of active nests	Proportion of successful nests until offtake	Proportion of fledglings surviving until offtake	Clutch size	Fledgling density
2002	Mean	4.43	0.40	0.58	3.944*	4.05
	Lower 95% CL	4.28	0.30	0.39	3.48	1.74
	Upper 95% CL	4.57	0.47	0.81	4.41	7.68
2003	Mean	4.22	0.67	0.59	3.944	6.56
	Lower 95% CL	4.11	0.61	0.43	3.48	3.75
	Upper 95% CL	4.32	0.68	0.80	4.41	10.37

*The value of clutch size was not estimated for 2002; the average 2003 clutch size was used (Mean = 3.944, Std. Dev. = 0.938, N = 18).

4.1.2.5 Estimation of fledgling density from the harvest area

Four people from the “*Proyecto Elé*” staff carried out a survey in the “*Impenetrable*” of the Chaco Province, from July 18 to July 27, 2004. This survey aimed at adding more information for the estimation of blue-fronted parrot fledgling density.

The objective of this campaign was to survey the active nest-trees of the 2003-2004 season (here “active nest-trees” implies those nests that had fledglings at the time of the 2004 harvest, i.e., did not include those nests that failed in producing fledglings). A total of 16 posts or houses that had been assigned a fledgling quota in the 2004 season were included, and all nest-trees from where fledglings were collected were surveyed. For each post the following information was recorded: the GPS location of all nest-trees source of an offtake, the number of the nests, the name of the collector or collectors, any particular characteristic of the nests, and in some instances the path followed by the collectors to get to the nest-trees. A verbal survey was also carried out with the collectors to get information about the search method used to locate the nests.

The results of this survey included information from 92 nest-trees. The average number of nest-trees per post was 5.75 (std. dev. = 1.44, N = 16, minimum 4 nest-trees, and maximum 10 nest-trees). As each nest-tree had its coordinates recorded with a GPS system, they were converted to a Cartesian coordinates system. The base area for estimating nest-tree density was 250 ha (2.5 km²), which is the property size of most collectors.

The nest-trees density remained, as expected, quite constant: an average of 2.30 nest-trees/km² with a coefficient of variation of 25.0 % (N= 16, lower 95% CL= 2.02 nest-trees/km², and upper 95% CL= 2.58; Table 22).

The mean number of fledglings/nest from the “*Loro Hablador*” Natural Reserve (average for the 2003-2004 season) is 2.36 fledglings/nest at offtake time. I preferred this value rather than the average number of fledglings per nest from the management area because the latter had to be corrected for the fledgling left in the nest, introducing some uncertainty (see Table 24). Using the “*Loro Hablador*” Natural Reserve fledglings/nest value we would be expecting a fledgling density of 5.43 fledglings/km² (N= 16, lower 95% CL= 4.40 nest-trees/km², and upper 95% CL= 6.58; Table 22).

Table 22: Average values, and their lower and upper 95% confidence limits, of nest-trees density (from the management area), fledglings/nest (from the “*Loro Hablador*” Natural Reserve), and the calculated fledgling density per km².

Statistic	Nest-trees/km ²	Fledglings/nest	Fledglings/km ²
Average	2.30	2.36	5.43
Lower 95% CL	2.02	2.18	4.40
Upper 95% CL	2.58	2.55	6.58

These density values of nest-trees/km² and fledglings are probably a clear subestimation, for the nest-trees selected by the collectors tend to be closely distributed as a result of the collector’s interest to have the fledgling offtake sites as nearby as possible to each other to save time and effort. This means that assigning the whole property surface (250 ha) as the area to be used for the calculation of density is excessive, thus subestimating fledgling density. The bias introduced by using the property size as the base surface for estimating densities is of the order of 36% lower in relation to the average density of nest-trees/km² obtained from the direct method in the “*Loro Hablador*” Natural Reserve (3.6 nest-trees/km²).

4.1.2.6 Delimiting the values of parameters r and K based upon actual offtake densities

One aspect of interest is to look at the behavior of the logistic model and the fixed escapement rule of management assuming that the population is at or near a stable level, and see if parameters r and K can be narrowed down to some more specific values. The reason to resort to this assumption is based on the relatively stable offtakes that took place in the last seven years in the “*Impenetrable*”. Table 23 shows the data of the “*Proyecto Elé*” activity from the season 1997/1998 to the last season (2003/2004) available. The surface area under offtake does not necessarily indicate the actual area searched for blue-fronted parrot fledglings, for each collector is assigned a quota and once this quota is fulfilled the collectors stop searching. Using this area as the best information available and knowing the total number of fledglings collected, the offtake population densities (in fledglings/km²) for each harvest season can be calculated (last column of Table 23). This value is probably an overestimation, for—as mentioned above—

most collectors use only a portion of their property to carry out their offtake, until their quota is fulfilled.

Table 23. Information on the “*Proyecto Elé*” activities from the season 1997/1998 to the last season (2003/2004) available. The surface area under management is the sum of the properties of all collectors (Source: “*Proyecto Elé*” internal reports).

Period	Surface area (km ²) under management	Collectors authorized	No. of effective Collectors	Quotas Granted	Fledglings Collected	Fledglings collected/km ²
1997/1998	169	56	45	350	312	1.85
1998/1999	384	109	69	700	589	1.53
1999/2000	584	132	98	920	661	1.13
2000/2001	712	171	129	1350	929	1.30
2001/2002	1070	270	186	1780	1393	1.30
2002/2003	1278	376	236	2180	1950	1.53
2003/2004	1464	450	312	2590	2412	1.65
Average=						1.47

When measuring offtake density in this way it is striking that the offtake shows to be relatively constant. It is true that in part this results from the “*Proyecto Elé*” managers’ strategy of trying to increase slowly the total offtake by expanding the area under management and not to increase the intensity of offtake in the same places (number of collectors and total quotas increase more or less in the same proportion as the surface area under management). However, assuming that the collector’s effort is relatively constant, this also suggests that the offtake seems to be maintained (and even seems to increase slightly in the last four years) at a level that apparently can be sustained by the blue-fronted parrot population. There are two caveats: (a) as in the “*Impenetrable*” the offtake is composed of only the youngest year class (fledglings), it may also be a case of unsustainable harvest, but that needs more than seven years to become evident, and (b) the offtake may be locally unsustainable but being replenished from surrounding unharvested areas.

With the aim of delimiting better the possible range of r and K values, I carried out a series of simulations applying the same offtake as the one actually carried out in the field from 1997/1998 to present (2003/2004) (i.e., an average of 1.47 fledglings/km²), assuming three possible scenarios: (a) that the population density has remained stable, (b) that the population density has declined 20% in those seven years, and (c) that the population density has increased 20% in those seven years. Then we can ask ourselves the question: which are the r and K values compatible with the above assumptions (a)-(c) in the presence of the an average annual offtake of 1.47 fledglings/km²?

In order to answer this question we would have to approximate to what might be the average population density in the management area. One of the most reliable data is the average density of nest-trees one estimated from the “*Loro Hablador*” Natural Reserve for the 2003-2004 season (2.6 nest-trees/km², with the direct method; see Table 13), and the best estimate of the mean number of fledglings per nest (3.08 fledglings/nest) was

obtained from a set of 481 nests inspected by the staff of “*Proyecto Elé*” in three seasons (Table 24).

Table 24. Number of fledglings per nest at the time of the offtake in the management area in the 2001/2002, 2002/2003, and 2003/2004 seasons (Source: “*Proyecto Elé*” internal reports).

Season	Average	Std. Desv.	N	Lower 95% Conf. Lim.	Upper 95% Conf. Lim.
2001-2002	2.99	0.76	72	2.81	3.16
2002-2003	2.99	0.77	135	2.85	3.12
2003-2004	3.15	0.83	274	3.05	3.25
All seasons	3.08	0.81	481	3.01	3.15

The result of the product between 2.6 nest-trees/km² and 3.08 fledglings per nest at the time of the offtake is 8.0 fledglings/km². This value conforms quite well with the 2003/2004 fledgling density in the “*Loro Habrador*” Natural Reserve using adaptive sampling (6.56 fledglings/km², 95% CL: 3.75 and 10.37 fledglings/km², see Table 22), although larger than the density by the direct method (4.02 fledglings/km², 95% CL: 0 and 13.52 fledglings/km², see Table 14).

Using densities between 4 and 12 fledglings/km², the Solver procedure of the Excel spreadsheet was used to find the r and K values that would keep the *A. aestiva* fledgling population stable, or growing or declining by 20%, under an annual offtake of 1.47 fledglings/km². The Solver procedure was applied with the restrictions that K should not be larger than 27 fledglings/km² (the largest value estimated for the fledgling density, in the management area; see Table 22), and that r should not be larger than 0.46 (the largest value estimated for r , with the Lande equation; see Table 25). Fig. 9 summarizes the results. We can see that, as expected from the logistic model, to sustain during seven sequential years an annual offtake of 1.47 fledglings/km² there are a series of combinations of different r and K values (each of the lines of Fig. 9) for any given level of fledgling density to be maintained (the exception was a fledgling density of 3 fledglings/km², that could not satisfy any of the (a), (b) or (c) conditions).

The general trend is that to sustain a larger fledgling density forces a combination of larger K and smaller r values. In Fig. 9 I have outlined with a dashed circle the possible combinations of r and K where most of the fledgling densities converge, particularly from densities of 7 or more fledglings/km². Thus, accepting that the blue-fronted parrot population might be in a stable state, our parameter values have been narrowed down to an r value of around 0.3 and K values between 18 and 22 fledglings/km². Assuming a 20% decrease in seven years, the parameter values would be narrowed down to an r value of slightly above 0.3 and K values between 14 and 18 fledglings/km², and assuming a 20% increase in seven years, the parameter values would be narrowed down to an r value of slightly below 0.3 and K values between 22 and 26 fledglings/km².

The results of this exercise in trying to delimit as much as possible the values of parameters r and K based upon actual offtake densities have to be taken with some reservations, for the actual offtake area of individual collectors is quite variable around

the average property size of 250 ha (see Table 23). Additionally, sometimes collectors fulfill their quota without covering completely their property, and in some other cases it is known that they collect fledglings from areas outside their properties. Thus the density values in the harvest area is only an approximation to the real offtake density.

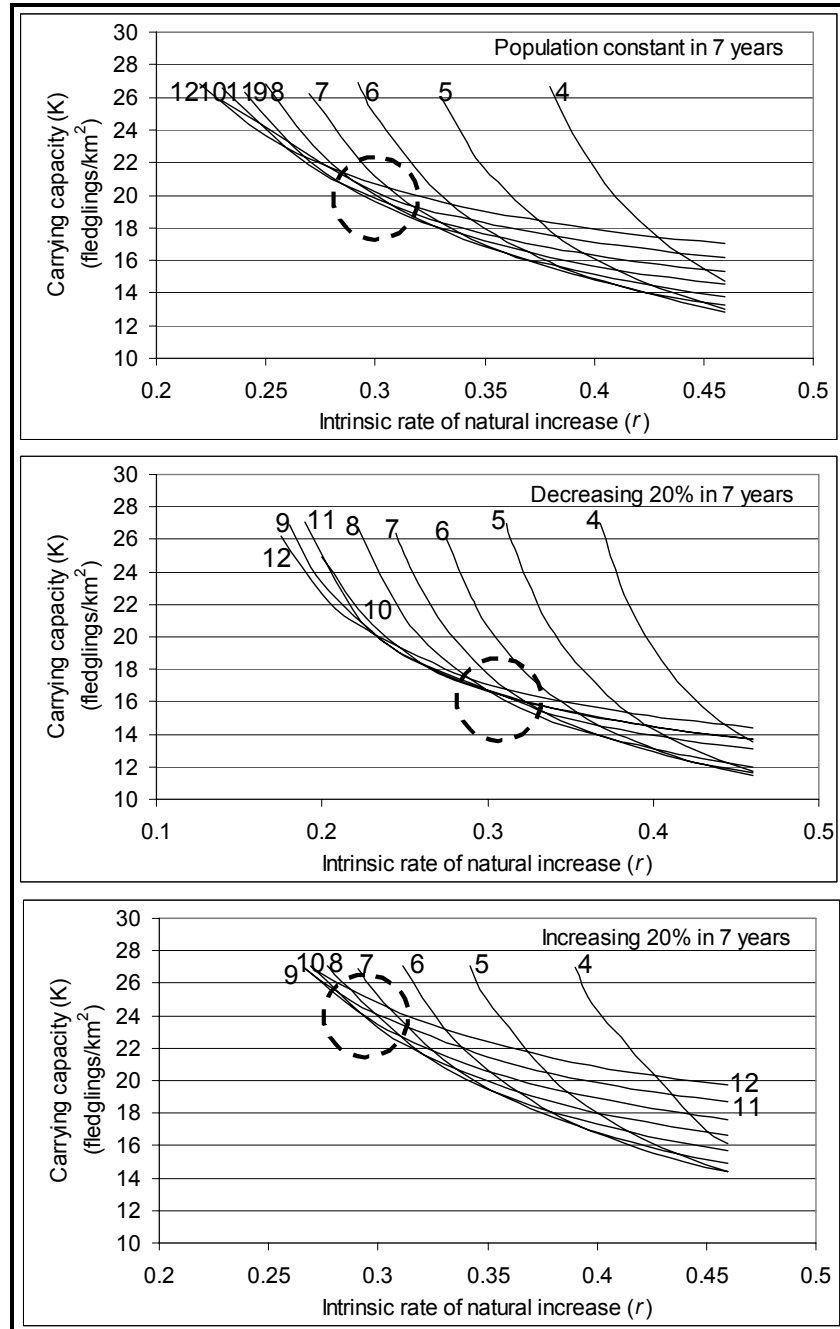


Fig. 9. Combinations of r and K that allow a constant or increasing or decreasing blue-fronted parrot population by 20%, during seven sequential years exposed to an annual offtake of 1.47 fledglings/km². The lines correspond to different fledgling density level. The dashed circles embrace the r and K combinations where stable population of 7 or more fledglings/km² seem to converge.

4.1.3 Summary of parameters' estimations by different methods

4.1.3.1 Summary of the intrinsic rate of natural increase (r)

With respect to the estimation of the intrinsic rate of natural increase (r), using the most plausible combination of parameters: (i) age of first reproduction (α): 2, 4 and 6 years, (ii) "skipping rates" 0 and 1, and survivorship curves: types II, III, I-II, Sig-A, Sig-B, and BM-80, the different estimations of r came out to be quite similar, suggesting a robust estimation. The r values estimated with Cole's formula (around 0.3) are slightly higher when compared to the ones estimated with Lotka's equation (range 0.12-0.18), and with the estimation of r using generation time and replacement rate (range 0.15-0.2). The homeotherm meta-analysis and the two ages matrix method also resulted in r values within the same range of Cole's formula (0.3). Thus we can conclude that an intrinsic rate of natural increase (r) in the range of 0.1-0.4 is an extremely plausible one and a conservative range to be used in a management simulation model. As the r values were estimated without a confidence region (this could be done by Jackknife or Bootstrap methods), and in order to avoid committing a possible sub-estimation, r was extended to values of up to 0.6 in the model simulations. Table 25 shows these results to facilitate comparison of the resulting r values among different methods.

Table 25. Range of estimated values of the intrinsic rate of natural increase (r) from the five methods used. For each method the range is the result of the most plausible "skipping rate" and age of first reproduction (α).

Estimation method	Range of r values produced
Lotka's equation	0.12-0.18
Cole's formula	0.31-0.32
Generation time and replacement rate	0.15-0.20
Homeotherm meta-analysis	0.3
Two ages matrix	0.34-0.46
Lande's equation	0.23-0.28

Fig. 10 allows a visual comparison of the range of results of the estimates of the intrinsic rate of natural increase (r) based upon the six methods used, and also let us identify the lower and upper limits of r under the criterion that they should include of the most credible results of all five methods, even if one of them is represented by one of its extreme values. Under such criterion the lower and upper limits of r are 0.12 and 0.46, respectively.

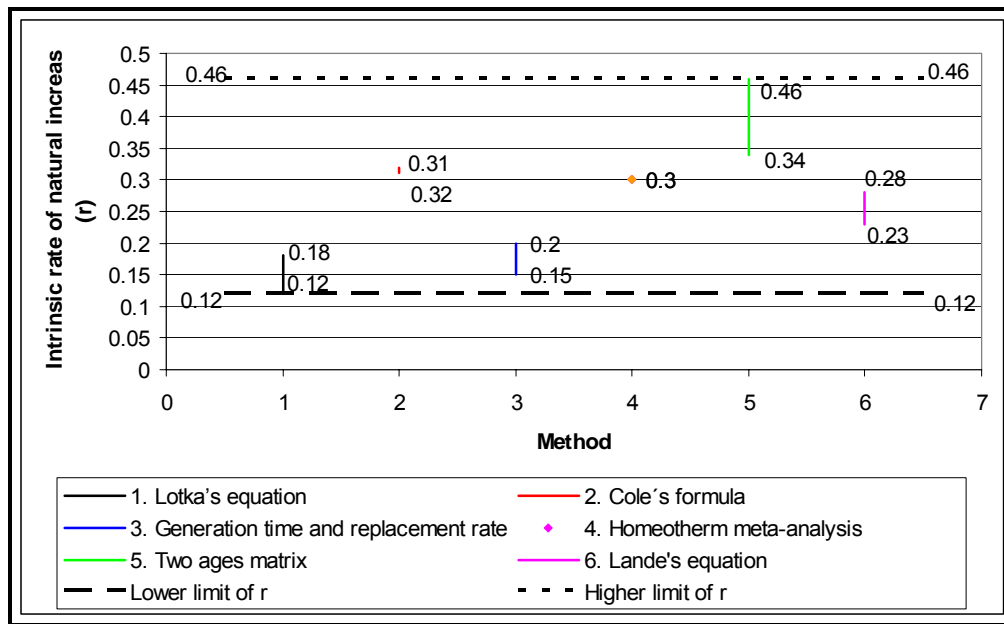


Fig. 10. Visual comparison of the results of the six methods used to estimate the intrinsic rate of natural increase (r) of *A. aestiva*. The black horizontal dashed lines represent the lower (0.12) and the upper (0.46) limits of r that result from including all estimation methods, even if they embrace the extreme of the range of a given method.

4.1.3.2 Summary of the carrying capacity (K)

The estimate of the fledglings density for a given year was considered a baseline for the estimation of the carrying capacity of the environment for fledglings of the blue-fronted parrot. Table 26 shows a summary of fledgling mean density by different methods and their respective 95% confidence intervals.

Without considering the value with the negative lower 95% confidence limit, fledgling density in the “*Loro Hablador*” Natural Reserve oscillates between about 0.1 to 6.5 fledglings/km² in 2002, and between 1.1 to 3.8 fledglings/km² in 2003. Averaging the means and 95% confidence limits of the different methods for each of these two periods we would have a fledgling density of around 5.3 fledglings/km² (95% CL: 2.1-9.2) for 2002 and 7.3 fledglings/km² (95% CL: 2.7-14.5) for 2003. Applying the same averaging procedure to the fledgling density results, without considering the Byth and Ripley method (the least reliable because of a sample size of only 6 random points), we would have a fledgling density of around 6.1 fledglings/km² (95% CL: 2.5-11.0) for 2002 and 8.0 fledglings/km² (95% CL: 3.1-15.5) for 2003. This procedure of averaging different methods does not have statistical grounds, and it was carried out just to see the order of magnitude of fledgling density in the “*Loro Hablador*” Natural Reserve.

Table 26. Summary of the blue-fronted parrot mean density (fledglings/km²) estimated by different methods and their respective 95% confidence intervals. The direct method is an estimate that assumes a surface of transects as a strip 500 m wide (Table 14). The transect method is based on independent sub-transects, using the Fourier model of detection with distance from the transects, and an estimation of the variance by the Jackknife procedure (Table 17). Byth and Ripley, and T-square are two Distance methods (Table 19). The adaptive sampling is a procedure to sample from clustered populations (Table 21). The fledgling density in the management area was presented in Table 22.

Place and time	Method	Fledgling mean density	Lower 95% CL	Upper 95% CL
"Loro Hablador" Reserve 2002-2003	Direct method	3.22	0.12	11.41
	Transect	12.67	6.51	18.82
	Byth & Ripley	1.94	0.49	2.05
	T-square	4.56	1.49	6.23
	Adaptive sampling	4.05	1.74	7.68
"Loro Hablador" Reserve 2003-2004	Direct method	4.02	-0.59	13.52
	Transect	17.13	7.33	26.94
	Byth & Ripley	4.44	1.13	10.44
	T-square	4.45	1.84	11.07
	Adaptive sampling	6.56	3.75	10.37
Management area July 2004	Direct method*	2.30	2.02	2.58

* In this case the direct method was applied based upon assigning a surface area of 2.5 km² to each of the 16 properties in the posts assigned an offtake quota (see Section 4.1.2.5).

The values of the 2003 sampling period are more reliable for this is the only sampling period where clutch size was measured in the field, and unless the population is already at the carrying capacity level, we would expect that carrying capacity should be somewhere above the 2003 fledgling density estimates. When we consider the highest fledgling density value as estimated from the management area (26.6 fledglings/km²), we can assign a range of possible K values between 2 and 26 fledglings/km² to the "Impenetrable" for the blue-fronted parrot, to use in the simulation and management models. Although the value of 26 fledglings/km² sounds as a high density estimate, it may be considered reasonable, for it should be noted that the upper 95% CL by the Transect method is 27 fledglings/km², and that this method is possibly the most reliable because of the independence of the Fourier series to the actual dispersion pattern of the population and to the reliability of the Jackknife method used to estimate variances. So the upper K value of 26 fledglings/km² used in the simulation and management models sounds plausible. Within this range of K from 2 to 26 fledglings/km² I consider, as a best guess, that a K value of around 20 fledglings/km² may be the most credible one.

In summary, from the results of the parameters' estimates I used six values of the intrinsic rate of natural increase ($r = 0.1, 0.2, 0.3, 0.4, 0.5$, and 0.6), and 13 values of the

carrying capacity ($K = 2, \dots, 26$ fledglings/km², with steps of 2). Rounding numbers the intrinsic rate of natural increase (r) was stretched one point over the maximum estimated (0.46 rounded to 0.5) just to cover the possibility that this parameter might in the higher end of the values estimated.

4.2 The Logistic Population Growth Model

Fig. 11 shows the behavior of the logistic population growth model for a selected series of values of r and K , assuming no stochasticity in the carrying capacity.

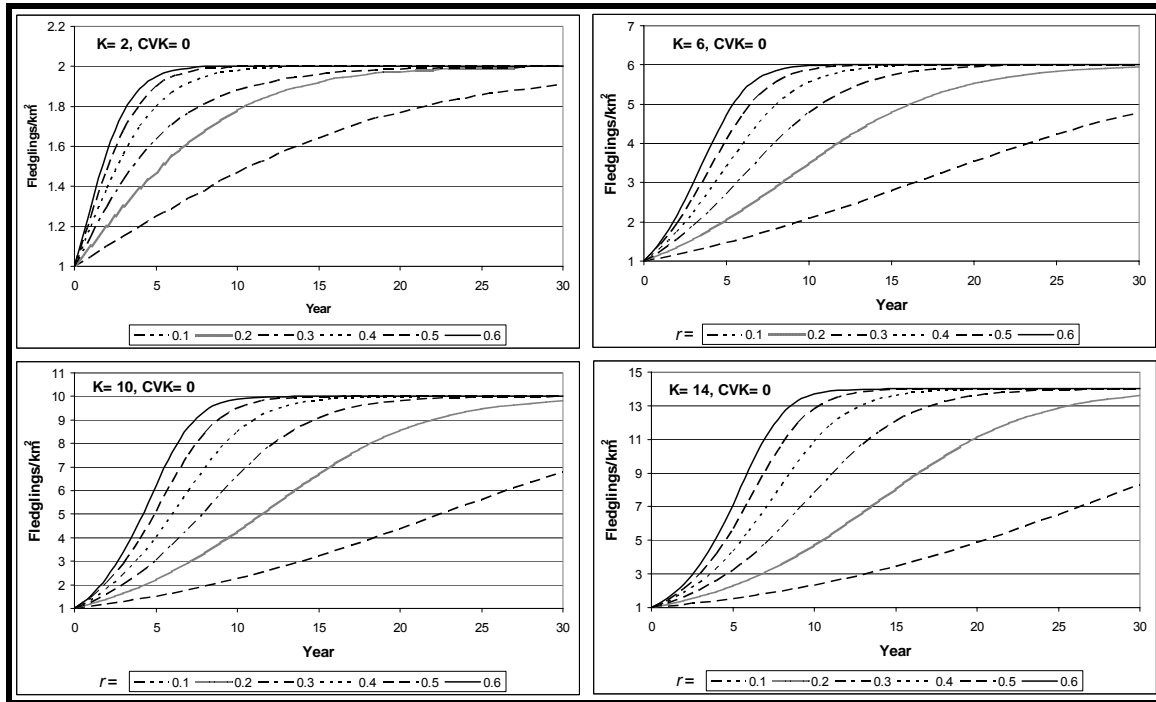


Fig. 11. Behavior of the logistic population growth model for 30 years simulations, for some of the most plausible r and K values, assuming no stochasticity in the carrying capacity.

The effects of the stochasticity of the carrying capacity (CVK) can be observed in Fig. 12, for a selected pair of r and K values ($r = 0.3$ and $K = 10$). Three levels of CVK were used: zero, 20 and 40 % of the coefficient of variation of the K value, assuming a normal distribution.

Only one of the K variability levels (CVK = 20%) was included in Fig. 12, just to illustrate the values it assumes from the random number generator, which can be greater or smaller than the average K value of 10. The parallel shape of the population curves for CVK 20 and 40% in Fig. 12 results from the fact that I used the same “seed” to generate a random number in all simulations. Despite the fluctuations of K during approximately the first 7-8 years of simulation the density of fledglings seems to be unaffected by this variation; this is because when the density is very low, the effects of K are minimal. It can

also be observed that the higher the stochasticity in K the lower the average density of fledglings.

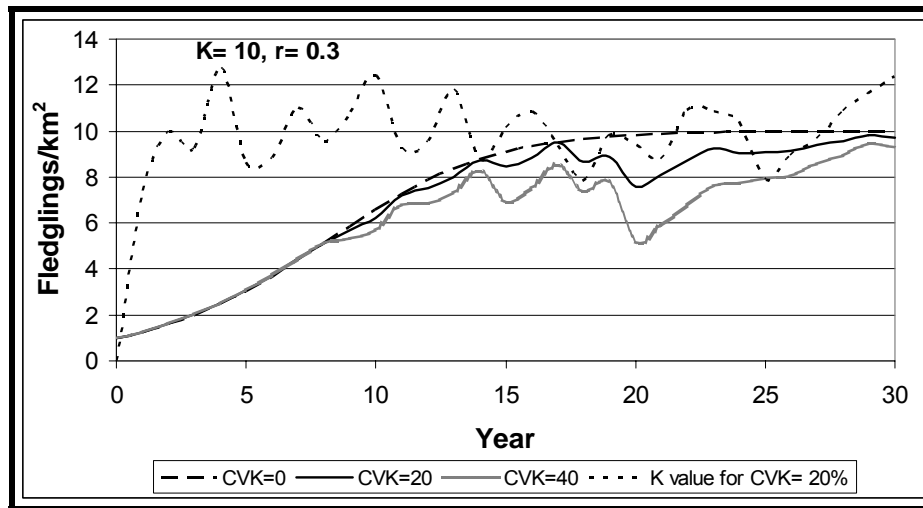


Fig. 12. Fledgling density for 30 years of the logistic simulation model for a selected pair of r and K values, and three levels of the coefficient of variation of K (0, 20 and 40 %). One simulated K value was also graphed (from CVK=20%), to illustrate the values K calculated by the random number generator.

4.3 The “fixed escapement” rule of population management

4.3.1 Behavior of population density under the “fixed escapement” rule of population management

Fig. 13 shows the fledgling density under management with the “fixed escapement” offtake rule, for selected values of r (0.3), K (10) and CVK (0, 20, and 40%). Fig. 13 also includes the potential fledgling density with no offtake as a reference level, as well as the offtake itself. It can be seen that the fledgling density stabilizes at about 4 fledglings/km² when CVK is zero and 20%, but drops to 3.7 fledglings/km² when CVK is 40%.

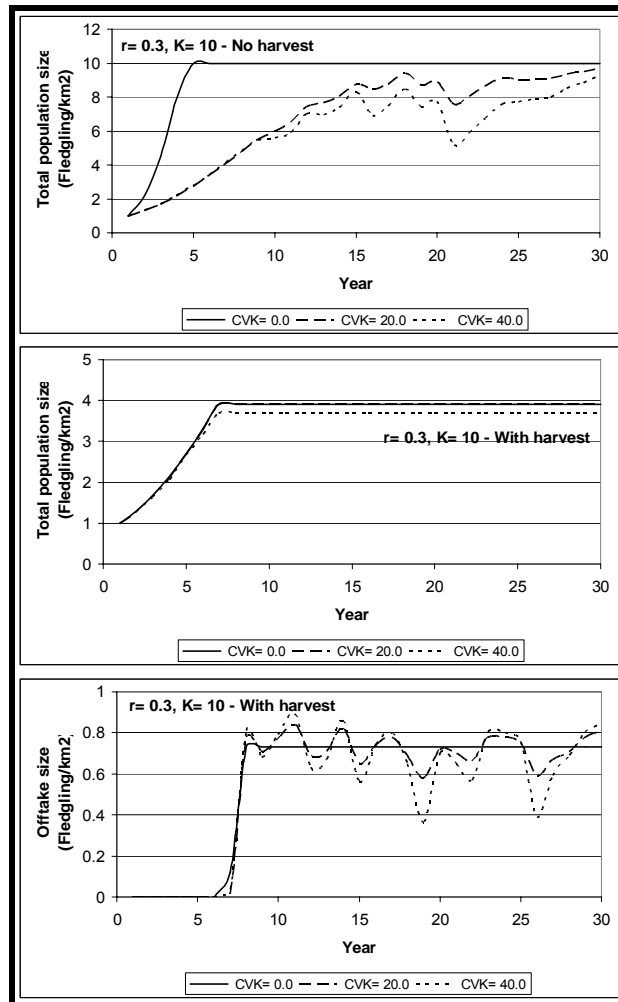


Fig. 13. Fledgling density under management with the “fixed escapement” offtake rule, for $r = 0.3$, $K = 10$ and $CVK = 0$, 20, and 40%. The potential fledgling density with no offtake and the offtake itself are included.

4.3.2 The threshold density under the “fixed escapement” rule of population management

The application of the “fixed escapement” management rule to the logistic population growth model during simulations was used to select the optimal escapement density, i.e., the threshold fledgling density that maximizes the accumulated offtake of fledglings during 30 simulated years. Table 27 shows the selected threshold densities that resulted for different values of the r and K parameters of the logistic population growth model, and for three levels of the stochasticity of the carrying capacity (K) expressed as its percent coefficient of variation (CVK).

Table 27. Optimal sustainable escapement (threshold) densities (in fledglings/km²) for various values of r and K of the logistic population growth model, and for three levels of the stochasticity of K , expressed as its percent coefficient of variation (CVK). Shaded cells are the most plausible combinations of r and K , and the figures in bold are considered the most plausible values.

Coefficient of variation of the carrying capacity (CVK)= 0 %						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.60	0.76	0.80	0.81	0.81	0.81
4	0.99	1.46	1.57	1.61	1.61	1.61
6	1.29	2.12	2.34	2.40	2.40	2.40
8	1.51	2.74	3.09	3.18	3.18	3.18
10	1.70	3.34	3.82	3.96	3.96	3.96
12	1.80	3.91	4.59	4.75	4.75	4.75
14	1.99	4.56	5.30	5.50	5.55	5.55
16	2.02	5.06	6.05	6.29	6.31	6.28
18	2.23	5.7	6.73	7.08	7.1	7.04
20	2.25	6.12	7.48	7.81	7.89	7.82
22	2.27	6.73	8.23	8.59	8.67	8.6
24	2.29	7.19	8.85	9.37	9.47	9.38
26	2.48	7.63	9.6	10.00	10.00	10.00
Coefficient of variation of the carrying capacity (CVK)= 20 %						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.59	0.74	0.79	0.80	0.80	0.78
4	0.97	1.44	1.56	1.58	1.57	1.57
6	1.28	2.09	2.31	2.37	2.35	2.32
8	1.46	2.67	3.05	3.16	3.14	3.10
10	1.65	3.26	3.76	3.90	3.93	3.88
12	1.80	3.88	4.51	4.69	4.72	4.65
14	1.99	4.40	5.15	5.44	5.51	5.43
16	2.02	4.92	5.89	6.22	6.23	6.22
18	2.14	5.53	6.62	6.99	7.01	6.99
20	2.25	5.93	7.29	7.68	7.79	7.77
22	2.27	6.51	8.02	8.45	8.57	8.54
24	2.29	7.1	8.75	9.22	9.35	9.32
26	2.38	7.35	9.36	9.99	10.00	10.00
Coefficient of variation of the carrying capacity (CVK)= 40 %						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.57	0.71	0.76	0.77	0.77	0.76
4	0.95	1.38	1.50	1.51	1.51	1.52
6	1.21	1.99	2.21	2.29	2.26	2.24
8	1.44	2.56	2.93	3.06	3.04	2.98
10	1.61	3.15	3.59	3.76	3.82	3.76
12	1.78	3.69	4.31	4.52	4.59	4.51
14	1.87	4.11	4.92	5.24	5.35	5.27
16	2.01	4.7	5.57	5.99	6.03	6.06
18	2.04	5.2	6.27	6.74	6.78	6.81
20	2.19	5.78	6.94	7.48	7.53	7.57
22	2.27	6.09	7.61	8.11	8.29	8.33
24	2.28	6.65	8.3	8.84	9.04	9.09
26	2.3	7.2	8.99	9.58	9.79	9.84

Fig. 14 shows the same information as Table 27 but in a graphical way. The contour lines of the escapement density (fledglings/km²), as a function of the intrinsic rate of natural increase (r) and the carrying capacity (K), for each level of variability of K , are relatively horizontal (practically no change above $r=0.3$), indicating a larger influence of K rather than r in determining the escapement density.

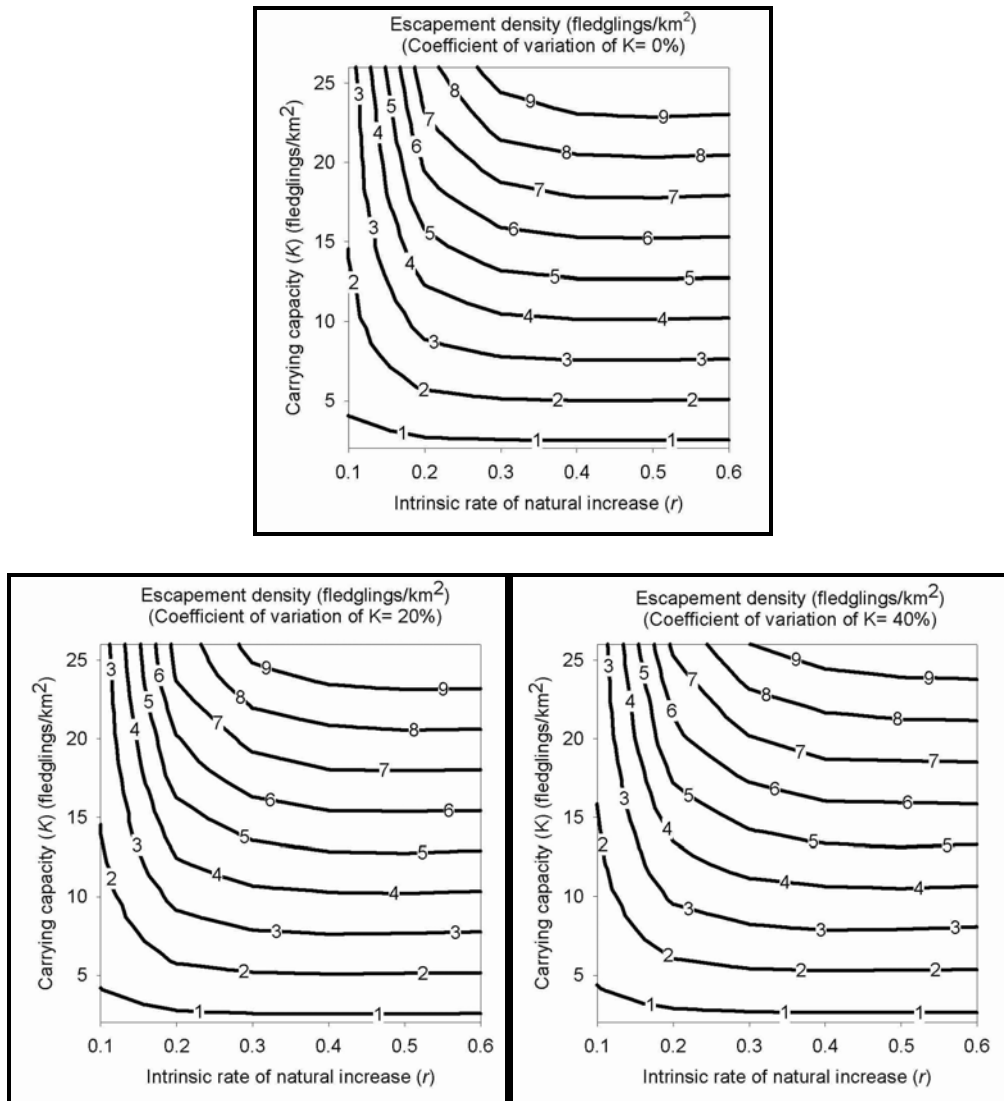


Fig. 14. Contour lines of the escapement density (fledglings/km²) as a function of the intrinsic rate of natural increase (r) and the carrying capacity (K), for three levels of variability of K . The isolines are relatively horizontal (practically no change above $r=0.3$), indicating a larger influence of K than r in determining the escapement density.

4.3.3 The optimal offtake density under the “fixed escapement” rule of population management

Table 28 shows the interaction between the intrinsic rate of natural increase (r) and the carrying capacity (K) in terms of offtake density (fledglings/km²), for three levels of

variation of K , in terms fledgling density offtake. Fig. 15 shows the same results as total number of fledglings harvested from a surface of 1464 km^2 , that represents approximately the summation of the properties of the collectors that were assigned a permit and a quota.

Table 28. Optimal sustainable offtake (fledglings/ km^2) for the various combinations of r and K , for three levels of variation of K . Shaded and bold cells as in Table 27.

Variation in $K=0\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.04	0.10	0.15	0.20	0.25	0.30
4	0.08	0.19	0.29	0.39	0.49	0.59
6	0.10	0.28	0.44	0.59	0.74	0.89
8	0.13	0.37	0.58	0.79	0.99	1.19
10	0.14	0.46	0.73	0.98	1.23	1.48
12	0.15	0.54	0.87	1.18	1.48	1.78
14	0.15	0.63	1.02	1.38	1.73	2.07
16	0.16	0.70	1.16	1.57	1.97	2.37
18	0.16	0.76	1.3	1.77	2.22	2.67
20	0.17	0.81	1.45	1.96	2.47	2.96
22	0.17	0.86	1.59	2.16	2.71	3.26
24	0.18	0.91	1.73	2.36	2.96	3.56
26	0.18	0.96	1.88	2.55	3.19	3.84
Variation in $K=20\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.04	0.09	0.14	0.19	0.24	0.29
4	0.07	0.18	0.28	0.38	0.48	0.57
6	0.10	0.27	0.43	0.57	0.72	0.86
8	0.12	0.36	0.57	0.76	0.96	1.15
10	0.14	0.44	0.71	0.95	1.20	1.44
12	0.15	0.53	0.85	1.14	1.44	1.72
14	0.15	0.61	0.98	1.33	1.67	2.01
16	0.16	0.69	1.12	1.52	1.91	2.3
18	0.16	0.75	1.26	1.72	2.15	2.58
20	0.17	0.80	1.40	1.90	2.39	2.87
22	0.17	0.85	1.54	2.09	2.63	3.16
24	0.17	0.90	1.68	2.28	2.87	3.44
26	0.18	0.95	1.82	2.47	3.10	3.73
Variation in $K=40\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	0.04	0.09	0.13	0.18	0.23	0.27
4	0.07	0.17	0.27	0.36	0.45	0.54
6	0.10	0.26	0.40	0.54	0.68	0.81
8	0.12	0.34	0.53	0.72	0.90	1.08
10	0.14	0.42	0.66	0.90	1.13	1.35
12	0.14	0.50	0.80	1.08	1.35	1.63
14	0.15	0.57	0.93	1.26	1.58	1.90
16	0.16	0.69	1.12	1.52	1.91	2.3
18	0.16	0.75	1.26	1.72	2.15	2.58
20	0.17	0.80	1.40	1.90	2.39	2.87
22	0.17	0.85	1.54	2.09	2.63	3.16
24	0.17	0.90	1.68	2.28	2.87	3.44
26	0.18	0.95	1.82	2.47	3.10	3.73

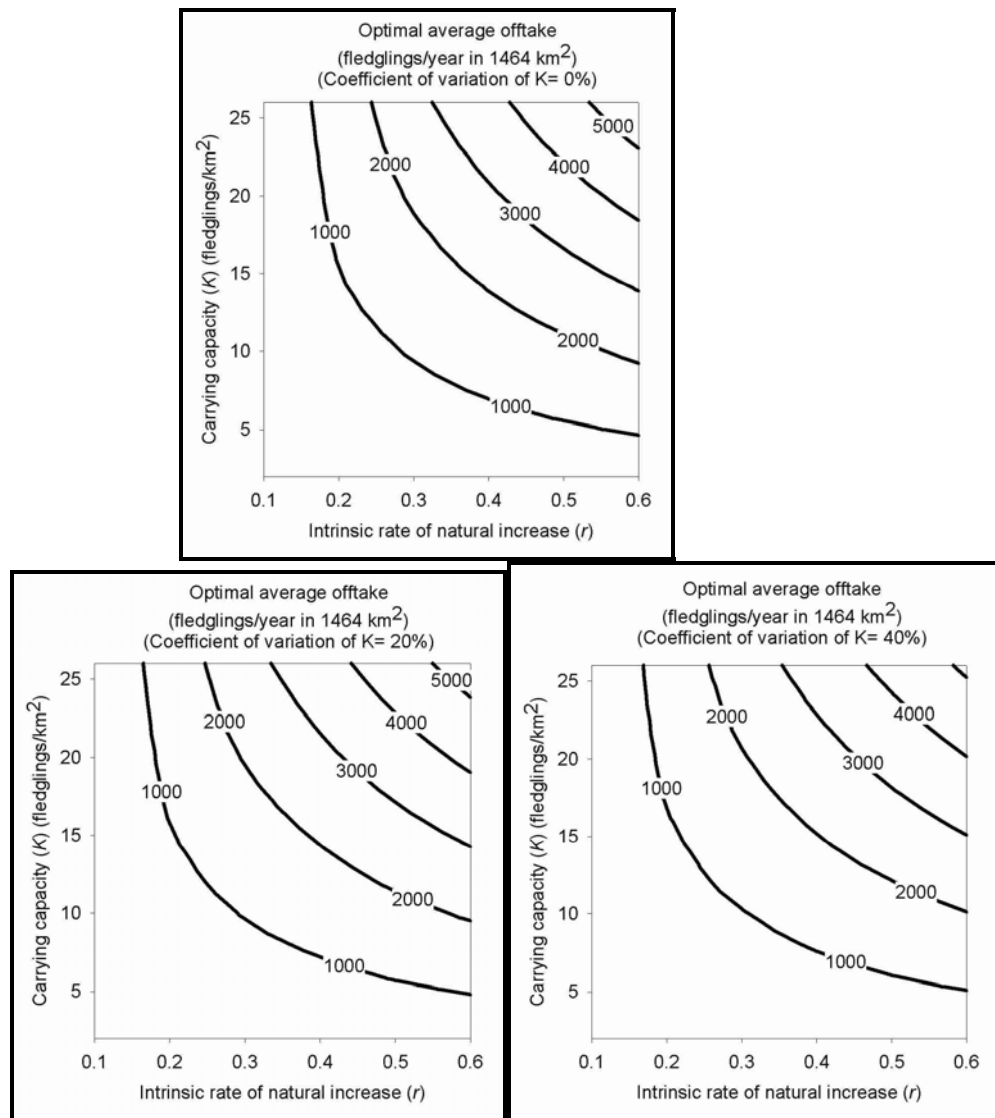


Fig. 15. Optimal annual offtakes of fledglings of the blue-fronted parrots from the “Impenetrable” Dry Chaco area of the Province of Chaco, as determined by the logistic simulation model under the “fixed escapement” management rule, for different combinations of the carrying capacity (K , in fledglings/km²), the intrinsic rate of natural increase (r), and three levels of the coefficient of variation of K . The offtakes were calculated as the optimal sustainable offtake and expressed as fledglings/1,464 km², an area that represents approximately the summation of the property size of the collectors that were assigned a quota in 2003/2004.

4.3.4 The optimal population offtake under the “fixed escapement” rule of population management

Table 29 shows the same results as Table 28, but the optimal sustainable offtake is expressed in total *number* of fledglings harvested from a surface of 20,000 km², that represents the total area of the “Impenetrable” under management.

Table 29. Optimal sustainable offtake in total fledglings/20,000 km² for the various combinations of the intrinsic rate of natural increase (r) and the carrying capacity (K), for three levels of variation of K . For an explanation of the shaded cells and values in bold see text.

Variation in $K=0\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	856	1925	2946	3953	4945	5932
4	1527	3798	5868	7899	9889	11864
6	2083	5630	8789	11836	14834	17789
8	2527	7415	11690	15764	19762	23721
10	2807	9172	14571	19692	24690	29637
12	2968	10887	17492	23629	29626	35569
14	3100	12699	20350	27528	34571	41493
16	3207	14030	23250	31465	39481	47426
18	3298	15177	26074	35401	44417	53333
20	3378	16261	28973	39280	49353	59257
22	3445	17294	31873	43207	54281	65181
24	3503	18293	34623	47134	59226	71105
26	3554	19230	37523	50903	63899	76876
Variation in $K=20\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	834	1861	2855	3828	4793	5743
4	1487	3690	5696	7643	9568	11491
6	2048	5472	8522	11464	14348	17216
8	2441	7179	11338	15285	19135	22959
10	2786	8888	14127	19070	23923	28701
12	2948	10628	16950	22892	28710	34438
14	3078	12246	19667	26683	33497	40181
16	3188	13806	22481	30497	38236	45930
18	3278	14928	25285	34303	43017	51666
20	3358	16001	28026	38041	47797	57409
22	3427	17010	30830	41847	52577	63146
24	3486	17987	33633	45653	57358	68888
26	3537	18920	36306	49459	62037	74566
Variation in $K=40\%$						
Intrinsic rate of natural increase (r)						
K	0.1	0.2	0.3	0.4	0.5	0.6
2	792	1751	2685	3602	4515	5423
4	1429	3474	5359	7189	9017	10845
6	1927	5145	8012	10797	13523	16252
8	2371	6771	10671	14399	18044	21667
10	2727	8413	13283	17963	22563	27099
12	2895	9998	15941	21560	27079	32518
14	3026	11433	18503	25132	31590	37941
16	3138	13070	21096	28724	36063	43374
18	3231	14368	23736	32315	40569	48793
20	3310	15403	26347	35900	45074	54216
22	3382	16385	28958	39407	49586	59638
24	3443	17316	31589	42984	54092	65061
26	3497	18221	34219	46569	58597	70480

When seen on this scale the potential total sustainable offtake is much higher than the present harvests (1393, 1950, and 2412 fledglings harvested in the 2001/2002, 2002/2003, and 2003/2004 offtake seasons, respectively). The shaded values are the optimal sustainable offtakes that result from the most plausible values of r and K (with the value in bold being the most credible one). The upper left cells of Table 29 with bold numbers and a diagonally shaded pattern identify those combinations of r and K that are below in approximately the lower 10% of the actual offtake in the “*Impenetrable*” in 2003-4 (2653 fledglings); i.e., those are the combinations of r and K that imply that present offtakes (2412 fledglings in 2003-4) are above the optimum sustainable offtake; i.e., they may be sustainable but non-optimal. However, it can be seen that they correspond to the combinations of the most lower r and K values estimated, which are not deemed very plausible. On the other hand it can be seen that the potential (maximum) *sustainable* offtake of the blue-fronted parrot would be in the order of 16,000-40,000 fledglings/year (shaded area in Table 29), being about 29,000 fledglings/year the most credible value.

The effects of the environmental stochasticity, as expressed by the increasing levels of the coefficient of variation in K (the three “sections” of Tables 28 and 29) do not seem to produce a dramatic effect in the levels of total sustainable offtake for the whole area of the “*Impenetrable*”. In terms of what it may be called offtake elasticities (the relative change of the offtake per unit change in the level of the coefficient of variation of K) it can be seen that the elasticities are low: around 5% decrease in the offtake for a change from 0 to 20% in the coefficient of variation of K , and around 11% decrease in the offtake for a change from 0 to 40% in the coefficient of variation of K .

4.3.5 Offtake as a proportion of the carrying capacity (K) density

Table 30 shows that the population of blue-fronted parrot fledglings are stabilized at densities that are various fractions of the carrying capacity (K). However, as the intrinsic rate of natural increase (r) becomes larger, the logistic model and the fixed escapement rule predict that the fledgling population tends to stabilize at approximately 50% of the carrying capacity, at least for low K values. This conforms the theoretical predictions, for it is known that the maximum population growth in the logistic population model occurs at densities of $K/2$.

Finally Table 31 also shows offtakes (as in Table 28) but expressed as a proportion of the carrying capacity. This unit conversion was of interest because it shows that, under the logistic model and the fixed escapement rule, offtakes turn out to be a relatively constant proportion of the carrying capacity for each value of the intrinsic rate of natural increase (r). The relationship between the offtakes as a proportion of the carrying capacity and the intrinsic rate of natural increase (r) is a linear one (regressions for all K values were statistically significant with $p < 0.05$) with an average slope of 0.25. That is, as a rule of thumb, we can estimate offtakes as one fourth of the r value in terms of a given value of K .

Table 30. Stable population density (escapement density) of fledglings (fledglings/km²) than can support optimal sustainable offtakes of Table 28 expressed as a proportion of its carrying capacity, after the logistic model and the “*fixed escapement*” rule of management, for different combinations of the intrinsic rate of natural increase (r) and carrying capacity (K , in number of fledglings/km²). Shaded cells are the most plausible combinations of r and K , and the value in bold the most credible value among them.

K	Intrinsic rate of natural increase (r)					
	0.1	0.2	0.3	0.4	0.5	0.6
2	0.30	0.38	0.40	0.41	0.40	0.40
4	0.25	0.37	0.39	0.40	0.40	0.40
6	0.22	0.35	0.39	0.40	0.40	0.39
8	0.19	0.34	0.39	0.40	0.40	0.39
10	0.17	0.33	0.38	0.40	0.40	0.39
12	0.15	0.33	0.38	0.40	0.40	0.39
14	0.14	0.33	0.38	0.39	0.40	0.39
16	0.13	0.32	0.38	0.39	0.39	0.39
18	0.12	0.32	0.37	0.39	0.39	0.39
20	0.11	0.31	0.37	0.39	0.39	0.39
22	0.10	0.31	0.37	0.39	0.39	0.39
24	0.10	0.30	0.37	0.39	0.39	0.39
26	0.10	0.29	0.37	0.38	0.38	0.38

Table 31. Offtake density of fledglings (fledglings/km²) than can support optimal sustainable offtakes of Table 28 expressed as a proportion of its carrying capacity, after the logistic model and the “*fixed escapement*” rule of management, for different combinations of the intrinsic rate of natural increase (r) and carrying capacity (K , in number of fledglings/km²). Shaded cells are the most plausible combinations of r and K , and the value in bold the most credible value among them (in this case relatively constant for a given value of r).

K	Intrinsic rate of natural increase (r)					
	0.1	0.2	0.3	0.4	0.5	0.6
2	0.020	0.050	0.075	0.100	0.125	0.150
4	0.020	0.048	0.073	0.098	0.123	0.148
6	0.017	0.047	0.073	0.098	0.123	0.148
8	0.016	0.046	0.073	0.099	0.124	0.149
10	0.014	0.046	0.073	0.098	0.123	0.148
12	0.013	0.045	0.073	0.098	0.123	0.148
14	0.011	0.045	0.073	0.099	0.124	0.148
16	0.010	0.044	0.073	0.098	0.123	0.148
18	0.009	0.042	0.072	0.098	0.123	0.148
20	0.009	0.041	0.073	0.098	0.124	0.148
22	0.008	0.039	0.072	0.098	0.123	0.148
24	0.008	0.038	0.072	0.098	0.123	0.148
26	0.007	0.037	0.072	0.098	0.123	0.148

4.4 Summary of management results

Under the assumption that the most plausible values of the intrinsic rate of natural increase (r) are between 0.2 and 0.4, and those of the carrying capacity around 20 fledglings/km² then, for management purposes in the “*Impenetrable*”, maintaining the blue-fronted parrot fledgling population in stable densities of the order of 6-8 fledglings/km² results in an average *sustainable* offtake of the order of 0.8-1.96 fledglings/km².

5 Discussion

The estimation of the parameters necessary to apply the logistic model of population growth (the intrinsic rate of natural increase, r , and the carrying capacity, K) were not directly available for *A. aestiva*, so indirect methods of estimation had to be resorted to. Informal sensitivity analysis of the simulation results indicated that possible errors in the values of these parameter have relatively serious effects on the decision making process in terms of sustainable management. Thus, it is important to make an appraisal of the likelihood or, at least, of the plausibility of the parameter values obtained by the indirect methods here used.

5.1 The intrinsic rate of natural increase (r)

The estimation of the intrinsic rate of natural increase (r) with the Lotka equation required assumptions about the survival and fecundity age-specific schedules of the blue-fronted parrot under wild conditions, as well as some demographic parameters, such as the intermittence in reproduction and the offspring sex ratio.

5.1.1 Survivorship curves

Few bird species have survivorship curves information gathered from the field. One interesting exception in the Psittacidae is the Major Mitchell or pink Cockatoo (*C. leadbeateri*) that is found on the margin of the Western Australian wheatbelt (Rowley and Chapman, 1991). These authors found a survivorship curve similar to a type III curve, but it has to be noted that this species is a short-lived cockatoo (about 5 years of life span) with a longevity much shorter than the blue-fronted parrot. If we refer to Table 2, that relates the different survivorship curves to average life span, we see that from a perfect type III survivorship curve an average longevity of 6.6 years was expected.

Generalizations from the few species that have been studied has lead to the widely accepted knowledge that birds are classical example of Deevey's type II survivorship curve (see Ganter, 2004). The Binkley and Miller (1980) method adapted to the blue-fronted parrot, and based on its basic clutch size (which has good statistical estimates) and a broadly estimated adult population density estimated from nest densities, resulted in an l_x curve between type I and type II, but much closer to a type II survivorship curve (although slightly bended upwards at middle ages of the blue-fronted parrot). This

similarity becomes more evident looking at the average longevity that results from each curve (17.4 years from the type MB-80 l_x curve and 14.5 years from the type II l_x curve). The main difference between them is that the former has higher mortality rates at the younger ages of 2-10 years (see Fig. 7) than the type II l_x curve, which has relatively low and constant mortality rates until about the age of 20 years, and then starts to show a sharp increase. In both cases the dominance of a constant mortality rate with age is a common feature. Interesting enough, this characteristic of a constant mortality rate with age in a species with an average life span of 30 years yields an overall annual population mortality rate of about 0.966.

The plausibility of the MB-80 l_x curve is also corroborated by the regression analyses based upon weight and reach of a variety of bird species, which lead to a longevity estimate of about 22 years for the blue-fronted parrot. At first glance, correlations based upon a variety of different families of both tropical and temperate species would seem unsound; however Karr *et al.* (1990) found that survival rates of tropical and temperate birds are similar. Additionally, that the average longevity of the blue-fronted parrot must be around 15-18 years is also suggested by the generalization proposed by Lindstedt and Calder (1976). These authors, comparing the weight and the longevity of a variety of bird species, propose that there is a non-linear (allometric) relationship that follows the function $L = a(W)^b$, where L is longevity expressed in years, W is weight expressed in kg, and a and b take the values of 21.6 and 0.26, respectively, for passerine birds. After assigning a weight of 0.5 kg to the adult blue-fronted parrot, the application of Lindstedt and Carter's equation results in a longevity of 18 years.

These results are quite different from the ones obtained by Collazo *et al.* (2003) for the endangered species *A. ventralis*. These authors found an average annual survival value of 0.65 for the first year (95% confidence level: 0.42, 0.87), as compared with the 0.966 survival of the first year according to a type II l_x curve, or the 0.977 survival of the first year according to a MB-80 l_x curve. However, there are not many grounds for an adequate comparison, for not only *A. ventralis* is a smaller species (length of 28 a 29 cm) than *A. aestiva* (about 37 cm), but also the survival values reported by Collazo *et al.* (2003) were the result of the follow up of captive-reared birds oriented towards a release program. Thus not only the real age of the *A. ventralis* released was not known (or at least not informed), but also the survival rate of captive-reared birds is possibly very different than that from birds born in the wild. Sandercock *et al.* (2000) estimated the overall local survival rate (ϕ) of the Green-rumped Parrotlet (*Forpus passerinus*) in Venezuela, and obtained a value of $\phi = 0.565$ (although breeders had a significantly higher probability of local survival ($\phi = 0.678$) than non-breeders ($\phi = 0.486$). However, overall, males and females had comparable local survival rates (breeders $\phi = 0.698$ vs. 0.658, nonbreeders $\phi = 0.536$ vs. 0.436). The measure of overall survival rates (i.e., annual survival of individuals that represent a mix of many different ages within the population) may result so heterogeneous that Sandercock *et al.* (2000) question whether the analyses of mean rates are meaningful at all; they claim that different species could have the same mean survival rate, but have markedly different survivorship curves due to age-specific variation in survival.

5.1.2 The maternity function

The other piece of information necessary for the Lotka equation is the maternity function (the age-specific fecundity schedule). There is even less ground for an adequate and plausible estimate of this function than for the survival function. The clutch size estimate here used (3.944, Igor Berkunsky, *pers. comm.*) is quite reliable (std. dev. 0.938, $n = 18$), and similar to estimates of other parrot species, e.g., the Nanday Parakeet (*Nandayus nenday*) in Paraguay: based on 32 natural nests on palm trees of *Copernicia alba*, Morales and Vitale (1997) found that the average clutch size was 3.6 eggs per nest, and a reproductive rate of 1.43 chicks per reproductive pair. Although it is claimed that in captivity (under ideal rearing conditions) once the blue-fronted parrots start laying eggs they go on doing so in a relatively constant manner during the rest of their reproductive life (in regularity and in clutch size, see Appendix VI), there are few indications if this holds in the wild. Actually, on the contrary, captive-rearing experts claim that *A. aestiva* nesting pairs, even in captivity, are extremely sensitive to alterations of all sorts (sounds, smells, visual signals, predators), frequently abandoning the nest when any of those disturbances take place. We would guess that under wild conditions these kind of problems (flooding of the nests in extremely rainy seasons, or the presence of predators) must be relatively frequent. As only the clutch size was quite reliable, the only way to consider these uncertainties was to keep them as explicit variables in the maternity function.

One way to represent these uncertainties that may show up in the continuity of reproduction is by considering the blue-fronted parrot as an intermittent breeder. This is relatively common in birds (i.e., nonbreeding in individuals that previously bred) and seems to be a widespread phenomenon among nonbiennial species (Newton, 1985, Hatch, 1987, Boekelheide and Ainley, 1989, Hamer *et al.*, 1991). Breeding proportions in populations have been estimated as extensive in ducks (Coulson, 1984), in penguins (Williams and Rodwell, 1992), in petrels (Chastel *et al.*, 1993), and in shags (Aebischer and Wanless, 1992). At the population level, the proportion of potential breeders that actually breed affects the population growth rate, and the factors that influence this proportion can be important population regulation mechanisms (Hémery *et al.* 1986). Very few studies have focused on the relationship between nonbreeding and demographic parameters such as survival probability or future reproduction (Cam *et al.*, 1998).

The experience with the blue-fronted parrot in aviaries (Eugeni Fernández, *pers. comm.*, see Appendix VI) shows that, when well fed and healthy, *A. aestiva* can breed every year. We do not know to what degree this condition occurs in the wild, but possibly an extremely harsh summer (too dry, poor tree fructification, and thus poor bird condition) could lead to some degree of intermittence in breeding. Thus using an intermediate “skipping rate” of 1 and 2 seemed reasonable. I may recall (see discussion on carrying capacity below) that sometimes intermittence in breeding may be associated with the “distance” to the carrying capacity of the environment; as the population numbers get close to the carrying capacity in many species intermittence in breeding starts to take place.

5.1.3 Offspring sex ratio

Another implicit but an essential and directly impinging additional factor in the estimation of r is the offspring's sex-ratio, for the maternity function in Lotka's equation is based upon an exclusively female population. This is why I went to extra efforts to justify why I used a 1:1 progeny sex ratio (i.e., 50% males and 50% females) in the blue-fronted parrot despite very little information is available, either under wild or commercial rearing conditions. Although progeny sex ratio in the blue-fronted parrot apparently is strongly male-biased (up to 30 males:70 females ratios; Eugeni Fernández Nogales, *pers. comm.*; see Appendix VI), I decided to be cautious with this information for it originated from *A. aestiva* in captivity, under extremely favorable feeding and rearing conditions. The careful experience with the kakapo, and the two hard data case studies with field populations of *A. auropalliata* and *A. aestiva* (despite their low sampling numbers) strongly suggest that we should not expect a marked alteration in the 1:1 offspring sex ratio in the blue-fronted parrot. Having developed a full discussion of the problem of the progeny sex ratio in birds in general, and in parrots in particular (see Appendix VII) I will not deal further with this topic here.

5.1.4 Cole's formula, the replacement rate and generation time, and homeotherm meta-analysis

With respect to the use of Cole's formula to estimate the intrinsic rate of natural increase (r), the three main parameters are α , the age of first reproduction, b , the clutch size, and ω , the age of last reproduction. The first two parameters have already been discussed as part of the age-specific survival and fecundity schedules used in the Lotka's formula. In terms of the age of last reproduction (ω) its estimation is closely associated with the average longevity, for in a species as the blue-fronted parrot, with no important postreproductive life, ω should be quite similar to the average lifespan. This in turn, is also closely related to the age-specific survival curve (l_x) already discussed in relation to the Lotka's equation method of estimating r .

The use of the replacement rate and the generation time to estimate the intrinsic rate of natural increase (r) also resorts to the age-specific survival and fecundity schedules used in the Lotka's formula, so there is no need to go deeper in its discussion, except that its reliability should be taken with reservation, for it is usually used as a first approximation for solving the Lotka equation.

Finally the homeotherm meta-analysis of Fenchel (1974) based upon the weight of the blue-fronted parrot has little uncertainty, despite the usual variability of the sexes and the state of the individual weighed.

5.1.5 Comparison of the results from the six different methods

Accepting what I have called the "most plausible" parameter combinations (those most "credible" l_x and m_x curves, and the most "reasonable" value of α , ω , and "skipping

rate”), we end up with the following: 0.15 from the Lotka equation, 0.32 from Cole’s formula, 0.20 from the generation time and replacement rate method, 0.3 from the homeotherm regression analysis, 0.42 from the two ages matrix method, and 0.27 from the Lande equation. Only the Lotka equation and the replacement rate and generation time method use the same type of information in a different manner, but the other two methods (Cole’s formula and homeotherm meta-analysis) resort to independent information. Thus the six methods used to estimate the intrinsic rate of natural increase (r) have proved to be relatively robust. In summary, despite the originally wide range of potential estimates the intrinsic rate of natural increase (r), the six methods seem to point towards an r value between 0.2 and 0.4 for the blue-fronted parrot. This is confirmed by the possible r values in the management area, even supporting an average offtake of 1.47 fledglings/km²/year, that result in an approximate r value of 0.3, independently of the assumption that the population be in a stable condition or increasing or decreasing 20% in 7 years.

5.2 The carrying capacity

The results here presented show that the carrying capacity (K) of the “*Impenetrable*” for the blue-fronted parrot is as critical as the intrinsic rate of natural increase (r) in terms of sustainable population management of the blue-fronted parrot. This statement is not only valid for the logistic model of population growth, but also equally valid for whatever other population model would have been used. In simplistic terms the carrying capacity is the amount of blue-fronted parrots which the “*Impenetrable*” can support or “carry” at one given time. To this “classical” definition I should add that it refers to the number of parrots on a *sustained basis*. This amount depends on its turn on three components: amount and quality of the food, cover, and water. In other words, these are the habitat requirements which determine the carrying capacity. These, in turn, are determined by other factors, such as the climate, the kind of soil and its fertility and, of course, by the way the land is used by people.

Habitat requirements for the blue-fronted parrot change during the seasons of the year. The food they eat in the winter is quite different than what they eat in the summer. The diet of the blue-fronted parrot is relatively well known, as well as the plant species that provide natural food sources, either seasonally or year round. The general characteristics of the habitat and the type of cover required by the blue-fronted parrot are also relatively well known. However, the conversion of this information into a quantitative measure of carrying capacity is not trivial. For example, cover is a habitat requirement that not only is essential against predation, but also prevents waste of energy. Additionally there are many other components that affect carrying capacity *per se*, e.g., the amount of edge and size of continuous areas, the vertical structure or layering of the vegetation, the structure of the tree canopy, the health status of the population, its genetic structure, and many others. However, key to the conversion of habitat requirements into carrying capacity numbers, other elements are needed, mainly those related to the population dynamics of the blue-fronted parrot, and among them, those related to the population regulation of the blue-fronted parrot numbers. In other words, we need to learn about what are normally

called the “limiting factors”, i.e., those habitat requirements that may become in short supply, and prevent the blue-fronted parrot population from growing.

5.2.1 The limiting factors

The information available on the population dynamics of the blue-fronted parrot is weaker than the one available on its natural history. One essential question still to be answered refers to its main population regulation mechanism. Like most other parrot and hornbill species, *Amazona* species depend largely on cavities in trees for their nesting sites. So obviously one important “candidate” to constitute a “limiting factor” in the blue-fronted parrot is the number of trees with adequate nesting cavities. The other main “candidate” is the amount of food available to sustain a given blue-fronted parrot population. In terms of the former, Beissinger (2001) has suggested that for parrot species suitable cavities may be in short supply and limit parrots’ population size, reinforcing the generally accepted idea that nest sites are usually limited for cavity nesters, at least for temperate birds (Brawn and Balda 1988, Newton 1994, Martin and Li, 1992, Martin, 1996). This seems to be the case of the Thick-billed Parrots (*Rhynchopsitta pachyrhyncha*) in northwestern Mexico’s Sierra Madre Occidental, where nesting density was apparently related to availability of suitable cavities (Lanning and Shiflett, 1983). Marsden and Martin (1997) also claim that it seems likely that nest site availability in forest patches influences the local abundance of hole-nesting birds in Sumba, Indonesia; however, these authors found that the relationship was strongest in the large parrot species, but absent in the hornbill *Rhyticeros everetti* and the red-cheeked parrot *Geoffroyus geoffroyi*.

The availability of trees with adequate nesting cavities would have been the easiest way to quantify carrying capacity in the “*Impenetrable*”; however, after several years of management of the blue-fronted parrot in Argentina apparently this does not seem to be an important regulation mechanism. Despite there is some degree of logging in the area of the “*Impenetrable*”, informal appraisals (Igor Berkunsky, *pers. comm.*) as well as formal studies (Banchs, Moschione and Flombaum, 1996) indicate that frequently trees with what seems to be adequate nesting cavities are available and not used. Furthermore, tree nesting cavities that are apparently appropriate because they had been successfully used for nesting in one year are not used in other years, and then re-used again. A similar finding was obtained by Koenig (2001) in the Black-billed Parrot *Amazona agilis*, in Jamaica. In the “*Impenetrable*” it was found that 60% of nest cavities (std. dev. 37%, range= 0-100%, N= 38) were reused between 2002 and 2003 (Igor Berkunsky, *pers. comm.*); a relatively high value of 40% of no reuse of apparently adequate nest-cavities would suggest that this resource is not a “limiting factor” in the estimation of the carrying capacity of the “*Impenetrable*” for *A. aestiva*.

Apparently food availability (particularly in the critical dry season) may be a stronger candidate as a limiting factor. Its conversion into carrying capacity is difficult. One approach is to look into the energetics of the food and the energetics of the requirements of the blue-fronted parrots for survival and reproduction, i.e., an ecophysiological approach. The other is to try to correlate the population numbers in successive years with

the phenology of the vegetation (period and amount of flowering, fruiting, and seed production). Additionally this correlation can be related to climatic factors, to be able to anticipate years with higher or lower carrying capacities. This process is also a relatively intricate one, for usually there are many indirect associations and lags in the relationship between weather and vegetation phenology, as well as between vegetation phenology and bird population status and abundance. For the particular case of the blue-fronted parrot, the critical dry season is winter and most of the population moves to nearby riparian forested areas or migrates to the foothills of the yungas and nearby forests of citric fields. The reproductive period of *A. aestiva* (fledglings' offtake time in the Dry Chaco) is the wet season in the "*Impenetrable*" and food availability does not seem to be a limiting factor (except in years of a strong dry spell).

5.2.2 Indicators of the "distance" to the carrying capacity

Given the difficulty in estimating numerically the carrying capacity, there is also another alternative: to use an indicator of the "distance" between the population density at a given time and the carrying capacity of the environment. That is, as the population numbers get close to the carrying capacity, intraspecific competition becomes harsher, and some changes take place that, taken in combination, can be used as indicators that the population is approaching its carrying capacity. Some of these changes include one or more of the following:

- (A) *Ethological changes* (appearance of antagonistic behavior and/or displacements, an increase in the time invested in searching for food and not dedicated to offspring care, changes in the diet, reduction of the home range or territory, relaxation of fidelity to the nesting site).
- (B) *Individual responses* (physical condition –including body growth rate–, illness and parasitism incidence).
- (C) *Reproductive characteristics* (age of first reproduction, annual rate of reproduction of mature females, intermittence in breeding).
- (D) *Population aspects* (age structure, rate of survival –specially of juveniles–, degree of occupation of marginal areas, rate of change of population size).

To look for carrying capacity indicators implies the identification or a measure of some of the above changes, which is no simple task either. In the case of the blue-fronted parrot there are informal and circumstantial observations that may help to identify if some of those changes are taking place.

Ethological indicators like antagonistic behavior and/or displacements, or changes in the proportion of the time invested in searching for food caring offspring have not been recorded systematically in the "*Impenetrable*" blue-fronted parrot population. Changes in the diet are known between the summer and winter periods, but not the differences in successive years for the same season. Data related to home ranges or territories depend upon a telemetry study that is still in its planning phase.

Before egg-laying blue-fronted parrots are easily visible and there is an apparently fidelity to the nesting site that had been selected, and can be used as an ethological indicator of carrying capacity. This is substantiated by the fact that there is a high relationship between those initial sightings and the active nests some time later (Igor Berkunsky, *pers. comm.*, for *A. aestiva* and Koenig, 2001, for *A. agilis*). A similar finding was obtained for the Green-rumped parrotlet (*F. passerinus*), which showed a strong site fidelity of 95% (percentage of adults that moved within 500 m in consecutive years; Sandercock *et al.*, 2000). A weakening of the site fidelity would be expected in case the population would be reaching carrying capacity levels, which apparently is not happening at present in the “*Impenetrable*”.

There is scarce formal and systematic information about physical condition of the blue-fronted parrot in the “*Impenetrable*”, but at least they seem to be free of illness and parasitism incidence, because there are negligible deaths during stockpiling and transport of the chicks collected during the offtake. However, the feeding condition of the parrots has not been ascertained, and in dry years a certain fraction of the population may be undernourished.

I have already discussed some of the reproductive characteristics of the blue-fronted parrot and there is little field information on reproductive traits such as the age of first reproduction or some other indication as the annual rate of reproduction of mature females, or the intermittence in breeding, mainly because there is no idea of the age structure of the population. With respect to the juvenile rate of survival, although not quantified, apparently most of the chick deaths during brooding are due to external causes such as predation or nest flooding, and not internal causes such as malnutrition or illnesses. In this sense there would be another, although weak and indirect, suggestion that the blue-fronted parrot does not seem to be near its carrying capacity.

Blue-fronted parrots have not been observed to occupy marginal areas, that is, areas outside their normal range, as it would be expected if the main occupancy area would be heavily populated near carrying capacity.

With respect to the rate of change in population size as an indicator of the “distance” to carrying capacity levels there are no time series long enough as to provide a reliable trend. The constant offtake of the last seven years (see Table 23), relatively low as compared with the heavy offtake of the decade of the 80’s, suggests contradictory interpretations. On one hand it would seem that the offtake is compensated by the replacement by new generations. In the “*Impenetrable*” the blue-fronted parrot has a productivity rate of 1.35 (average number of fledglings produced alive per pair per year, as of the season 2003-2004; Igor Berkunsky, *pers. comm.*), and an overall annual survival rate (assuming a MB-80 type of survivorship curve) of 0.99 (see section “4.1.1.4. Two ages matrix approach”). Let us assume that there are 3 “active” nests/km² (an “active” nest is an adequate nest cavity that has clear evidences that it has been selected for reproduction by a blue-fronted parrot pair which show an incipient behavior of nest preparation or egg-laying activities) Let us assume that about 50% of the population reproduces annually (because of a certain proportion of adults do not reproduce every

year and because some juveniles do not start reproducing until later –the age of first reproduction–). Then 3 “active” nests/km² (this is the average between a density of 3.6 active nests for 2003, and of 2.6 active nests for 2004, at the “*Loro Hablador*” Natural Reserve; see Table 13) represents the presence of three pairs/km² or six individuals/km², and would correspond to a total adult population of 12 adult individuals/km² (6 individuals/km² divided by 0.5, where 0.5 is the fraction of the population that reproduces annually). Let us further assume that the overall annual mortality rate is of the order of 0.01 (see section “4.1.1.4. Two ages matrix approach” for its justification); this value although much lower than the overall annual mortality rate of about 0.35 obtained by Sandercock *et al.* (2000) for the Green-rumped Parrotlet (*Forpus passerinus*) in Venezuela, is somewhat smaller than the 0.05-0.10 overall mortality rate for second year (juveniles) and third year and onwards (adults) of *Amazona barbadensis*, used for a Population Viability Analysis by Rodríguez *et al.*, *in press*). Then about 0.12 parrots/km² (12 x 0.01) would die per year while about 4.1 parrots/km² would be added per year (3 pairs/km² x 1.35 fledglings produced alive per pair per year). This means a net surplus of about 4 parrots/km²/year.

So it would not be surprising that the blue-fronted parrot in the “*Impenetrable*” may be at (or near) a carrying capacity level. Unless, of course, that the offtake being applied in this area compensates the net surplus, and the population would be in a sort of balanced state between offtake and replacement, but below its actual carrying capacity. This is an open possibility, for it seems that during the last seven years of the “*Proyecto Elé*” management program, with an apparently constant effort per km², the offtake (also on a per km² basis) does not seem to have been diminishing.

5.2.3 Density as an approximation to the carrying capacity

Density estimates, particularly if obtained from a series of successive annual samplings, can be considered as a baseline for carrying capacities, except when those successive annual samplings show a permanent (increasing or decreasing) trend. The detection of aural signals when looking for and censusing nesting birds depends, for a given species, on its population density, the time of day, the stage in the breeding season, the habitat, and individual variation in behavior. Additionally, the probability that a bird will be detected by its song depends on how often the bird is on its territory, the frequency of singing, the length of time the observer is within potential detection distance, and the observer’s sensitivity. All of these factors contribute to an actual, observed frequency of detection that will seldom be equal to one (Emlen 1977; Diehl 1981, Christman, 1984).

The statistical problems related to a reliable sampling of fledglings or of an adult parrot population are not insurmountable, but they require a careful application of methods with verification of their main assumptions. In the case of the blue-fronted parrot no sampling independent from the management plans was carried out since the “*Proyecto Elé*” started, and the application of the nest and fledgling densities based solely upon the collectors’ data is not easily amenable to rigorous statistical analysis. However, although restricted to two years of data, the biological and ecological studies carried out at the “*Loro*

Hablador” Natural Reserve provide more reliable data to estimate densities, and these can be compared with a few estimates from other places and similar species.

Guix *et al.* (1999) censused four species of syntopic parrots using distance sampling methods on the 33,593 ha São Sebastião island, SE Brazil, in the protected area of Ilhabela Park, in a Mata Atlântica forest. These authors found that although the species counted had marked differences in size and weight, densities (individuals/km²) and estimated population size in 23,500 ha of well-preserved forests were similar: *Amazona farinosa* (13.82 ± 5.94; 3,247 ± 1,395), *Pionus maximiliani* (15.79 ± 7.04; 3,712 ± 1,654), *Brotoogeris tirica* (15.05 ± 4.87; 3,537 ± 1,143) and *Pyrrhura frontalis* (13.06 ± 5.53; 3,068 ± 1,298). Morales and Vitale (1997) found even greater densities in the Nanday Parakeet (*Nandayus nenday*) in Paraguay. These authors carried out studies on population trends and reproductive behavior in natural and artificial cavities of the Nanday Parakeet and estimated a population density of 1.2 individuals/ha (120 individuals/km²). Of course, *N. nenday* is a Parakeet much smaller than *A. aestiva*, with a length of only 30 cm and weighing only 140 g.

Average densities may be extremely variable depending on the species and the place. For example, Casagrande and Beissinger (1997), testing different sampling methods obtained population densities well over 200 individuals/km² for the Green-rumped Parrotlet in Venezuela. However, again as with *N. nenday* we must remember that the Green-rumped Parrotlet, in addition to being a much smaller bird than *A. aestiva* and an extremely fecund parrot species (Beissinger, 2001), it occupies a completely different habitat, climate and vegetation.

A digression may be in order here in relation to the age structure of the blue-fronted parrot population. I thought it would be worthwhile to compare the age-class proportion that we can approximate from field information, and the stable age distribution of the two ages matrix model. The main piece of information lacking from the field is the proportion of the adult population that breeds every year (what we had called c_a in the two ages matrix model). For that purpose I resorted again to the use of a range of values of c_a , and checked which of them resulted in a fledgling/adult proportion most similar to the results of the two ages matrix model. This process was carried out using the same information of the last two paragraphs of section 5.2.2 (3.06 “active” nests/km², 2.38 fledglings/km², and a certain number of blue-fronted parrot adults/km², depending on the assumption of what the c_a is). Table 32 shows the results of this exercise, illustrating that the proportions of fledglings in the “virtual field” population are always below the ones of the two ages matrix model, although this difference becomes smaller as the c_a values increase. However, this result suggests that either: (a) the blue-fronted parrot population of the “Loro *Hablador*” Natural Reserve probably has not reached a stable age distribution, or (b) the annual offtake of fledglings is distorting the age structure of the population.

The proportion of adult parrots that breed (c_a) is a demographic parameter seldomly estimated in the field. One of the few robust statistical estimate was obtained for the Green-rumped Parrotlet in Venezuela (Beissinger, 2001, Sandercock *et al.* 2000). Cam *et al.* (1998) analyzed the relationships between adult breeding state and the demographic

parameters “survival probability” and “subsequent breeding probability”, based on multistate capture–recapture models of the Kittiwake (*Rissa tridactyla*). Their conclusion was that (a) nonbreeding individuals have lower survival rates than breeders, (b) that nonbreeders have a higher probability of being nonbreeders the following year than do breeders, and (c) that nonbreeders tend to be lower quality individuals. However, it is not known if in the blue-fronted parrot the fraction of non-breeders in the population reflects poor individuals that may never breed, or if this is an indirect evidence of intermittence in breeding, genetically or energetically determined. Assuming that non-breeders in the blue-fronted parrot population is due to a biological intermittence, as a very general statement we could say that, if 50 % of the adult female parrot population are non-breeders at a given year but may become breeders the following year, then the “skipping rate” should be closer to 1 year (as about one half of the female population would be reproducing every year).

Table 32. Comparison of the proportion of fledglings in the population based on field densities and predicted by the Stable Age Distribution (SAD) of the two ages matrix model, as a function of the proportion of breeding adults (c_a). Except the last two rows with the proportions, figures are in number of individuals per km².

	Proportion of adults that breed (c_a)					
	0.5	0.6	0.7	0.8	0.9	1
Average “active” nests	3.06	3.06	3.06	3.06	3.06	3.06
Number of adults that breed	6.1	6.1	6.1	6.1	6.1	6.1
Total number of adults in population	12.2	10.2	8.7	7.7	6.8	6.1
Total number of fledglings in population	2.4	2.4	2.4	2.4	2.4	2.4
Total population (adults+ fledglings)	14.6	12.6	11.1	10.0	9.2	8.5
Proportion of fledglings in population	0.16	0.19	0.21	0.24	0.26	0.28
Proportion of fledglings predicted by the SAD of the two ages matrix model	0.30	0.33	0.35	0.38	0.40	0.41

5.3 The logistic model of population growth

The use of the logistic model of population growth can be considered as a first approximation for wildlife population management; however, although the management conclusions can be improved with more detailed models (such as matrix models with age specific survival and fecundity schedules and sex differences) this does not imply that the behavior of the logistic model should be disregarded as totally unrealistic or useless. On the contrary, Runge and Johnson (2002) have shown that the density-dependent logistic survival model has optimal equilibrium population size and maximum annual harvest intermediate between those usually called additive and compensatory models. Additionally, these authors claim that the logistic survival model does exhibit the property of compensation because increased harvest reduces the equilibrium population size.

5.4 The fixed escapement management rule

It has been widely accepted that harvesting may alter the dynamics of populations (Solberg *et al.*, 1999). In particular, when the population develops in a strongly fluctuating environment the effects on the dynamics may become quite complex. This complexity was first analyzed by Beddington and May (1977) and May *et al.* (1978).

Different harvest strategies have been applied for the management of wildlife populations. Constant harvesting involves removal of a fixed number of individuals. When the population is subject to proportional harvesting, a certain fraction of the population is removed. This is a commonly employed strategy for marine and terrestrial species (Getz and Haight 1979, Hilborn and Walters 1992), and it has been suggested to be particularly useful when the environment is strongly unstable (Hilborn and Walters 1992, Walters and Parma 1996, Quinn and Deriso 1999), which I think is the case of the blue-fronted parrot. Lande *et al.* (1995, 1997) concluded, however, that threshold harvesting, which I have called here “*fixed escapement*” harvesting (i.e., harvesting all individuals in excess of a threshold) was superior to those two other strategies for a number of different optimization criteria. In general, this strategy gives larger variance in yield than does proportional harvesting (Reed 1979, Clark 1985, Lande *et al.* 1995, 1997, Quinn and Deriso 1999).

These analytical results were derived under the assumption that quite accurate population and parameter estimates were available (Aanes *et al.*, 2002). However, a large number of studies, especially from marine systems, have documented that uncertainties in population projections should strongly affect the choice of harvesting strategy (Walters 1986, Hilborn and Walters 1992, Quinn and Deriso 1999). Reed (1979) and Aanes *et al.* (2002) claim that uncertain population estimates should lead to more conservative harvesting strategies, removing only a proportion of the difference between the population estimate and the threshold. Additionally, it has been shown (Aanes *et al.*, 2002) that restricted proportional harvest (there is only a fixed quota to harvest per unit time) resulted in slightly higher mean annual yields than proportional threshold harvesting (harvesting only a fixed proportion of the difference between the estimated population size and the threshold, when this difference is positive). However, variance in annual yield was reduced by restricted proportional harvesting because periods with low population size became shorter.

A caveat is in order in relation to the “*fixed escapement*” harvesting: it poses a strong demand on the robustness and reliability of the sampling schemes to estimate population density prior to the establishment of the harvesting quotas, as well as on the logistics of the management plan itself. Additionally the threshold harvesting management rule requires that the “*escapement*” population level be determined. Ideally this threshold may be estimated when a wildlife species has been managed in the field for some time, and the population response to harvests has been recorded and analyzed. However, in most species of commercial interest, as it is the case of the blue-fronted parrot, this information is not available, so more theoretical approaches for determining the “*escapement*”

population level had to be used, resulting in an additional dependence on the underlying population dynamic model and on the degree of reliability of the parameter estimates.

Beddington and May (1977) and May *et al.* (1978) using diffusion models showed, for a large variety of stock-recruitment curves, that exploitation may strongly reduce population stability. Thus, the dynamical consequences of harvesting should be considered when developing sustainable harvesting strategies. This has been done with the logistic model, and the effects on the population stability are well known (Clark, 1976), but the interesting properties in the case of the blue-fronted parrot will arise when a more realistic age and sex explicit model will be developed in a second phase of this analysis of sustainable management, and furthermore, when habitat-based metapopulation models are developed.

5.5 Some management considerations

Many parrot species are commercially valuable in the international pet trade and it has been suggested that conservation would be fostered if sustainable harvests would be in place (Thomsen and Brautigam, 1991, Beissinger and Bucher, 1992, Stoleson and Biesenger, 1997). However, sustainable use is a concept that does not carry the same meaning to everyone. I here adhere with the definition given by Beissinger and Bucher (1992): “the continued persistence and replenishment of a resource despite utilization”. But this persistence and replenishment is not only the art and science of manipulating wild populations for man's use, but also respond to the numerous philosophies as to what the managers (and the society at large behind them) really want in relation to a particular species population. Thus, with a particular population, the manager must employ various strategies to achieve the desired results in terms of numbers, age structure, sex ratios, condition levels, physical characteristics, and –of course– harvest rates.

There will be different opinions on what should be an adequate population level for a given habitat depending upon the management objectives for a particular species. For example, the management strategy will be different if the criterion is to manage for parrot numbers at the habitat's carrying capacity, than if the parrot population would be managed at a level of maximum sustainable yield. Populations managed at maximum sustainable yield will have fewer parrots than populations that are managed at carrying capacity. On the other hand, in the former case recruitment is higher (by recruitment I mean the number of parrots, born in the spring and summer, that survive long enough to be added to the population the next fall). However, regardless of each specific situation, society at large and managers in particular recognize that (with the exception of endangered species) one of the primary keys in managing any population is regulating the harvest.

One extremely important element that has been receiving increasing importance in wildlife management is the property rights aspect (Southgate, *et al.*, 1990, Steelman and Carmin, 1998, Burling, 1992, Bulte, 2002). I will not deal with this huge and complex topic here, but just mention that the “*Proyecto Elé*” is handling this problem by stimulating land acquisition (quotas are assigned only to those that have property titles or

can show that are in the process of obtaining them). This coincides with one of the prevalent concepts in wildlife management: that private property fosters taking care of the natural resources; however, there are also strong arguments against this position.

Of course when speaking of wildlife management of any kind, it is assumed that rules are followed as prescribed and that decisions are carried out with no errors. This is rarely so, and the legal and administrative aspects of managing wildlife become a real critical issue. Extreme cases of the importance of these aspects are those related to poaching, a common practice in many countries that have poor controls, low levels of vigilance and weak rule enforcement (Barnes *et al.*, 1993, Jachmann and Billiow, 1997, Trent, 1999).

For sustainable management the worst of all situations is when loose enforcement occurs in conjunction with a high level of corruption installed in the society (Milner-Gulland and Leader-Williams, 1992). In the present model neither the economics of the blue-fronted parrot sustainable management nor the poaching and corruption problems have been addressed, although these aspects have already been incorporated in some wildlife management mathematical models (Micheltore *et al.*, 1994). In the comparative analysis on poaching of parrot nests before and after the passage of the U.S. Wild Bird Conservation Act, carried out by Wright *et al.* (2001) on nest poaching in Neotropical parrots, the authors found that poaching rates were significantly lower following its enactment than in the period before. However, very justifiably, these authors warn that their results also show that the legal and illegal parrot trades are positively related, rather than inversely related as has been suggested by avicultural interests. The blue-fronted parrot management is no exception to the general rule of the difficulties of wildlife poaching control and the lack of data on wildlife poaching. Although the managers of the *A. aestiva* management program consider that poaching has diminished significantly since the present program has started, the area is known as a source of some illegal blue-fronted parrot fledglings, mainly for the domestic pet trade. However, accurate figures on poaching are lacking and future versions of this model will have to include the sensitivity of the optimal offtake threshold under the fixed escapement rule to different levels of possible poaching activity.

Finally, it can be argued, particularly in relation to many parrot species, that as there is a fraction of the adult population that are non-breeders, this may represent a surplus population amenable to harvesting without incidence on population sustainability; however this same factor reduces the rate of population growth (Beissinger, 2001). In the case of the blue-fronted parrot this would apply only to the mountainous rainforest areas of this species' range, for in the "*Impenetrable*" area we are dealing with, only fledglings are harvested. On the other hand, the offtake of chicks may be questioned as being of a higher impact on the parrot population. Having developed "virtual" life tables of the blue-fronted parrot, this question can have an approximate answer through the so called age-specific reproductive values.

Reproductive values represent the contribution of each age class to future populations. They can be computed from the age-specific survival and reproductive schedules, as well as from the projection matrix of the two ages matrix (they constitute the left eigenvector).

In a stable population that is neither increasing nor decreasing, reproductive value is defined as the age-specific expectation of future offspring. This is an important demographic concept, first elaborated by Fisher (1930), and commonly used in wildlife management and pest control. Reproductive values usually tend to peak at age of first reproduction (unless survival or fertility prospects continue to increase dramatically with age). For the blue-fronted parrot calculations with the type II survival curve the maximum reproductive value, as expected, peaked at the age of first reproduction (α), independently of the “skipping rate”. If we accept $\alpha=2$, then pre-reproductive juveniles of age 1 will have the same reproductive value as adults of age 3; if we accept $\alpha=4$, then pre-reproductive juveniles of ages 1, 2 and 3 will have the same reproductive value as adults of ages 5, 6 and 7; and so on. Thus, when comparing harvests of fledglings versus adults, the effective impact on the blue-fronted parrot population will depend upon the actual age of first reproduction (α) of that population as well as upon the age structure of the adult population.

6 Conclusions

It was determined that considering an intrinsic rate of natural increase (r) are between 0.2 and 0.4, and a carrying capacity around 20 fledglings/km² as the most plausible values of these parameters, then –for management purposes in the “*Impenetrable*” – if the blue-fronted parrot fledgling population is maintained in stable densities of the order of 6-8 fledglings/km², an average *sustainable* offtake of the order of 0.8-1.96 fledglings/km² can be obtained. Due to the high number of uncertainties in the parameters’ estimation we can say that a *sustainable* offtake between 1 and 2 fledglings/km² may be considered feasible.

It should be emphasized that these results, based upon the logistic model of population growth and the fixed escapement density management rule, represent an *optimal* sustainable offtake, but that many other levels of offtake may also be sustainable. The criterion for determining the optimal offtake values here obtained (between 1 and 2 fledglings/km², with a threshold or escapement density of 6-8 fledglings/km²) results from searching for the maximum accumulated offtake in a 30-years time span. Other criteria could be used (conservationist, economic or other) and the optimal sustainable offtake and threshold density would be different.

Acknowledgements

To Ricardo Banchs and Igor Berkunsky who provided invaluable data, bibliography and critical information, as well as excellent ideas while participating in group discussions. To Juan Carlos Reboreda who offered important suggestions and concepts. To Jon Paul Rodríguez of Fundación PROVITA (Caracas) for counseling on the bibliography of *A. barbadensis* in Venezuela and providing the example of a PVA study with this species. To Eugeni Fernández for aviary data and experience (Appendix VI). To Xavier Favoreux for ideas and opinions. To Roy Berger and Rick Jordan for the information provided by e-mail (Appendix V). To Waldo Hasperue for programming many of the sampling analyses and the Jackknife method.

Literature Cited

- Aanes, S., S. Engen, B.-E. Sæther, T. Willebrand and V. Marström. 2002. Sustainable harvesting strategies of willow ptarmigan in a fluctuating environment. *Ecological Applications* 12(1): 281-290.
- Aebischer, N. J., and S. Wanless, 1992. Relationships between colony size, adult non-breeding and environmental conditions for shag *Phalacrocorax aristotelis* on the Isle May, Scotland. *Bird Study* 39: 43-52.
- Anderson, D. R., J. L. Laake, B. R. Crain and K. P. Burnham 1976. Guidelines for Line Transect Sampling of Biological Populations. Utah Cooperative Wildlife Research Unit, Utah State University, Logan. 27 pp.
- Anderson, D. R., W. A. Link, D. H. Johnson and K. P. Burnham. 2001. Suggestions for presenting the results of data analyses. *J. Wildl. Manage.* 65: 373-378.
- Banchs, R. A., F. N. Moschione, and P. Flombaum. 1996. Características de los nidos de Loro Hablador y oferta de huecos en el Chaco central Argentino. IXth Argentinean Ornithological Meeting, Buenos Aires, October 14-16, 1996. Libro de Resúmenes, Asociación Ornitológica del Plata, Buenos Aires, Argentina, pp. 5-6.
- Banchs, R., F. Mochione, M. Codesido, P. Gado, and P. Grilli. 2000. Parámetros reproductivos de *Amazona aestiva* (Psittacidae) en el Chaco Argentino. *Ornitología Brasileira no Século XX*. Ediciones UNISUL.
- Barnes, R. F. W., M. Agnangna and M. P. T. Alers. 1993. Elephants and ivory poaching in the forests of equatorial Africa: report of a field reconnaissance. *Oryx* 27: 273-4.
- Basson, M., J. R. Beddington and F. R. M. May. 1991. An assessment of the maximum sustainable yield of ivory from African elephant populations. *Mathematical Bioscience* 104: 73-95.
- Beddington, J. R. and R. M. May. 1977. Harvesting populations in a randomly fluctuating environment. *Science* 197: 463-465.
- Beissinger, S. R. 2001. Trade of live wild birds: potentials, principles and practices of sustainable use. Pages 182-202. In: *Conservation of Exploited Species* (J. D. Reynolds, G. M. Mace, K. H. Redford, and J. G. Robinson, Eds.). Cambridge University Press, Cambridge, U.K.
- Beissinger, S. R. and E. H. Bucher. 1992. Can parrots be conserved through sustainable harvesting? *BioScience* 42: 164-173.

- Binkley, C. S. and R. S. Miller. 1980. Survivorship of the whooping crane, *Grus americana*. *Ecology* 61(2): 434-437.
- Boekelheide, R. J. and D. G. Ainley. 1989. Age, resource availability, and breeding effort in Brandt's cormorant. *Auk* 106: 389-401.
- Brauer, F. and D. A. Sánchez. 1975. Constant rate population harvesting: Equilibrium and stability. *Theoretical Population Biology* 8(1): 12-30.
- Brawn, J. D. and R. P. Balda. 1988. Population biology cavity nesters in northern Arizona: do nest sites limit breeding densities? *Condor* 90:61-71.
- Buckland, S. T. 1985. Perpendicular Distance Models for Line Transect Sampling. *Biometrics* 44: 177-195.
- Buckland, S.T., D. R. Anderson, K. P. Burnham and J. L. Laake. 1993. *Distance Sampling: Estimating Abundance of Biological Populations*. Chapman and Hall, London. 446 pp.
- Burling, J. S. 1992. Property Rights, Endangered Species, Wetlands, and other Critters, is it against Nature to Pay for a Taking? *Land and Water Review* 27(2): 312-362.
- Bulte, E. H. 2002. Open access harvesting of wildlife: the poaching pit and conservation of endangered species. *Agricultural Economics* 1657: 1-11.
- Burnham, K. P. and D. R. Anderson 1976. Mathematical Models for Nonparametric Inferences from Line Transect Data. *Biometrics* 32(2): 325-336.
- Burnham, K. P., D. R. Anderson and J. L. Laake 1980. Estimation of Density from Line Transect Sampling of Biological Populations. *Wildlife Monographs* No. 72. 202 pp.
- Byth, K. 1982. On robust distance-based intensity estimators. *Biometrics* 38:127-135
- Byth, K. and B. D. Ripley. 1980. On sampling spatial patterns by distance methods. *Biometrics* 36:279-284
- Cam, E., J. E. Hines, J-Y. Monnat, J. D. Nichols and E. Danchin. 1998. Are Adult Nonbreeders Prudent Parents? The Kittiwake Model. *Ecology* 79(8) : 2917-2930
- Casagrande, D. G. and S. R. Beissinger. 1997. Evaluation of four methods for estimating parrot population size. *The Condor* 99: 445-457.
- Caswell, H. 1978. A General Formula for the Sensitivity of Population Growth Rate to Changes in Life History Parameters. *Theoretical Population Biology* 14: 215-230.

Caswell, H. 1989. Matrix Population Models: Construction, Analysis, and Interpretation. Sinauer Associates, Inc., Sunderland, Mass., USA. 328 pp.

Chastel, O., H. Weimerskirch, and P. Jouventin. 1993. High annual variability in reproductive success and survival of an Antarctic seabird, the snow petrel *Pagodroma nivea*. *Oecologia* 94: 78–285.

Chaves, L. F., T. A. Revilla, D. J. Rodríguez and J. E. Rabinovich. 2004. Mortality profiles of *Rhodnius prolixus* (Heteroptera: Reduviidae), vector of Chagas disease. *Acta Tropica* (*in press*).

Christman, S. P. 1984. Plot Mapping: Estimating Densities of Breeding Bird Territories by Combining Spot Mapping and Transect Techniques. *The Condor* 86: 237-241.

Clark, C. W. 1976. Mathematical Bioeconomics: The Optimal Management of Renewable Resources. John Wiley & Sons, NY., USA. 352 pp.

Clark, C. W. 1985. Bioeconomic modelling and fisheries management. Wiley-Interscience, New York, New York, USA.

Cole, L. C. 1954. The Population Consequences of Life History Phenomena. *Quarterly Review of Biology* 29(2): 103-137.

Collazo, J. A., T. H. White Jr., F. J. Vilella and S. A. Guerrero. 2003. Survival of Captive-Reared Hispaniolan Parrots Released in Parque Nacional del Este, Dominican Republic. *The Condor* 105:198–207.

Comfort, A. 1962. Survival curves of some birds in the London Zoo. *Ibis* 104: 115-117.

Coulson, J. C. 1984. The population dynamics of the Eider Duck *Somateria mollissima* and evidence of extensive nonbreeding by adult ducks. *Ibis* 126: 525–543.

Deevey, E. S. Jr. 1947. Life tables for natural populations. *Quarterly Review of Biology* 22: 283-314.

Diehl, B. 1981. Bird populations consist of individuals differing in many respects, p. 225-229. In C. J. Ralph and J. M. Scott [eds.], Estimating the numbers terrestrial birds. *Stud. Avian Biol.* 6.

Diggle, P.J. 1983. Statistical Analysis of spatial point patterns. Academic Press, London.

Eberhardt, L. 1967. Some Developments in "Distance Sampling". *Biometrics* 23(2): 207-216.

Emlen, J. T. 1977. Estimating breeding season bird densities from transect counts. *Auk* 94: 455-468.

Fenchel, T. 1974. Intrinsic rate of natural increase: The relationship with body size. *Oecologia* 14: 317-326.

Fernandes Seixas, G. H. and G. de Miranda Mourão. 2002. Nesting success and hatching survival of the Blue-fronted Amazon (*Amazona aestiva*) in the Pantanal of Mato Grosso do Sul, Brazil. *J. Field Ornithol.* 73(4): 399-409.

Fisher, R. A. 1930. *The Genetical Theory of Natural Selection*. Oxford at the Clarendon Press, UK. 272 pp.

Ganter, P. 2004. Biology Department, Harned Hall 320, Tennessee State University, <http://www.tnstate.edu/ganter/B412%20Ch%206%20PopGrowth.html>, Internet site accessed June 12, 2004. Verification address: pganter@tnstate.edu

Gates, C. E., W. H. Marshall and D. P. Olson 1968. Line Transect Method for Estimating Grouse Population Densities. *Biometrics* 24: 135.

Gates, C. E. 1979. Line Transect and Related Issues. Pp. 71-14. IN: *Sampling Biological Populations* (R. M. Cormack, G. P. Patil & D. S. Robson, Editors). Statistical Ecology Series, 5. International Cooperation Publishing House, Maryland, USA. 392 pp.

Getz, W. M. and R. G. Haight. 1979. *Population harvesting*. Princeton University Press, Princeton, New Jersey, USA.

Gray, H. L. and W. R. Schucany. 1972. *The generalized jackknife statistic*. Marcel Dekker, Inc. New York, N. Y. 308 pp.

Guix, J. C., M. Martín and S. Mañosa. 1999. Conservation status of parrot populations in an Atlantic rainforest area of southeastern Brazil. *Biodiversity and Conservation* 8: 1079-1088.

Hamer, K. C., R. W. Furness, and W. G. Caldow. 1991. The effect of changes in food availability on the breeding ecology of great skuas *Catharacta skua* in Shetland. *Journal of Zoology*, London 223: 175-188.

Hatch, S. C. 1987. Adult survival and productivity of northern fulmars in Alaska. *Condor* 89: 685-696.

Heinsohn, R. and S. Legge. 2003. Breeding biology of the reverse-dichromatic, cooperative parrot, *Eclectus roratus*. *Journal of Zoology* 259: 197-208.

Hémery, G., E. d'Elbée, and J.-F. Terrasse. 1986. Régulation d'une population de pétrels-tempête *Hydrobates pelagicus* par reproduction intermittente. *Comptes-Rendus de l'Académie des Sciences*, Paris 303: 353-356.

- Hilborn, R. and C. J. Walters 1992. Quantitative Fisheries Stock Assessment. Choice, Dynamics and Uncertainty. Chapman and Hall. New York and London. 570 pp.
- Hines, W. G. S. and R. J. T. Hines. 1979. The Eberhardt index and the detection of non-randomness of spatial point distributions. *Biometrika* 66:73-80
- Hiraldo, F., J. J. Negro, J. A. Donazar and P. Gaona. 1996. A demographic model for a population of the endangered Lesser Kestrel in southern Spain. *Journal of Applied Ecology* 33: 1085-1093.
- Hopkins, B. 1954. A new method of determining the type of distribution of plant individuals. *Ann. Bot.* 18:213-227
- Hutchinson, G. E. and E. S. Deevey, Jr. 1949. Ecological Studies on Populations. *Survey Biological Progress* 1: 325.
- Hunter, C. M., H. Moller and D. Fletcher. 2000. Parameter uncertainty and elasticity analyses of a population model: setting research priorities for shearwaters. *Ecological Modelling* 134: 299-323.
- Jachmann, H. and M. Billiouw 1997. Elephant Poaching and Law Enforcement in the Central Luangwa Valley, Zambia. *Journal of Applied Ecology* 34(1): 233-244.
- Karr, J. R., J. D. Nichols, M. K. Klimkiewicz and J. D. Brawn. 1990. Survival rates of birds of tropical and temperate forests: will the dogma survive? *American Naturalist* 136: 277-291.
- Kendall, M. and A. Stuart. 1977. *The Advanced Theory of Statistics. Vol. 1. Distribution Theory.* 4th. Edition. Macmillan, New York, USA.
- Keyfitz, N. 1968. *Introduction to the Mathematics of Population.* Addison-Wesley Publishing Company, California, USA. 450 pp.
- Koenig, S. E. 2001. The breeding biology of Black-billed Parrot *Amazona agilis* and Yellow-billed Parrot *Amazona collaria* in Cockpit Country, Jamaica. *Bird Conservation International* (2001) 11:205-225.
- Krebs, C.J. 1989. *Ecological methodology.* Harper & Row: NY
- Kroon, H. de, A. Plaisier, J. van Groenendael and H. Caswell. 1986. Elasticity: the Relative Contribution of Demographic Parameters to Population Growth Rate *Ecology* 67(5): 1427-1431.
- Kroon, H. de, J. van Groenendael and J. Ehrlén. 2000. Elasticities: a Review of Methods and Model Limitations *Ecology* 81(3): 607-618.

- Laake, J. L., K. P. Burnham and D. R. Anderson 1979. User's Manual for Program TRANSECT. Utah State University Press. Logan, Utah. 26 pp.
- Lack, D. 1966. Population studies of birds. Clarendon Press, Oxford, England.
- Lancia, R. A., C. E. Braun, M. W. Collopy, R. D. Dueser, J. G. Kie, C. J. Martinka, J. D. Nichols, T. D. Nudds, W. R. Porath, and N. G. Tilghman. 1996. ARM! for the future: adaptive resource management in the wildlife profession. Wildlife Society Bulletin 24: 436–442.
- Lande, R. 1988. Demographics models of the northern spotted owl (*Strix occidentalis caurina*). Oecologia 75: 601–607.
- Lande, R., S. Engen and B.-E. Sæther. 1995. Optimal harvesting of fluctuating populations with a risk of extinction. American Naturalist 145:728–745.
- Lande, R., B.-E. Sæther and S. Engen. 1997. Threshold harvesting for sustainability of fluctuating resources. Ecology 78:1341–1350.
- Lande, R., B.-E. Sæther and S. Engen. 2001. Sustainable exploitation of fluctuating populations. Pp. 3-15. In: Conservation of exploited species. J. D. Reynolds, G. M. Mace, K. H. Redford and J. G. Robinson (Edit.). Cambridge University Press, Cambridge, UK. 524 pp.
- Lanning, D. V. and J. T. Shiflett. 1983. Nesting Ecology of Thick-Billed Parrots. The Condor 85: 66-73.
- Lindstedt, S. L. and W. A. Calder. 1976. Body size and longevity in birds. The Condor 78: 91-94.
- Lotka, A. J. 1922. The Stability of the Normal Age Distribution. Proceedings of the National Academy of Science 8: 339-345.
- Ludwig, D. and C. J. Walters 1981. Measurement Errors and Uncertainty in Parameter Estimates for Stock and Recruitment. Canadian Journal of Fisheries and Aquatic Sciences 38: 711-720.
- Marsden, S. J. and J. J. Martin. 1997. The Nesting Requirements of the Parrots and Hornbill of Sumba, Indonesia. Biological Conservation 82: 279-287.
- Marsh, T. L., F. G. Huffaker and G. E. Long. 2000. Optimal Control of Vector-Virus-Plant Interactions: The Case of Potato Leafroll Virus Net Necrosis. American Journal of Agricultural Economics 82(3): 556-569.
- Marshall, H. 1947. The longevity of the American herring gull. Auk 64: 188-198.

- Martin, T. E. 1996. Life history evolution in tropical and south temperate birds: what do we really know? *Journal of Avian Biology* 27:263–272.
- Martin, T. E. and P. Li. 1992. Life-history traits of open- vs. cavity-nesting birds. *Ecology* 73:579–592.
- May, R. M., J. R. Beddington, J. W. Horwood and J. Sheperd. 1978. Exploiting natural populations in an uncertain world. *Mathematical Biosciences* 42:219–252.
- McCallum, H. 2000. Population parameters. Estimation for Ecological Models. Blackwell Science Ltd., Oxford, UK. 348 pp.
- Michelmore, F., K. Beardsley, R. F. W. Barnes and I. Douglas-Hamilton 1994. A model illustrating the changes in elephant numbers caused by poaching. *African Journal of Ecology* 32: 89–99.
- Miller, R. G. 1974. The jackknife – a review. *Biometrika* 61(1): 1-15.
- Milner-Gulland, E. J. and N. Leader-Williams 1992. A model of incentives for the illegal exploitation of black rhinos and elephants: poaching pays in Luangwa Valley, Zambia. *Journal of Applied Ecology* 29(2): 388-401.
- Morales, C. and C. Vitale. 1997. Bases Biológicas para el Uso Sustentable del *Nandayus nenday* en Paraguay. Resúmenes presentados para el III Congreso Internacional sobre Manejo de Fauna Silvestre en la Amazonia y Latinoamérica. Santa Cruz, Bolivia. 3-7 de Diciembre de 1997.
- Newton, I. 1985. Lifetime reproductive output of female sparrowhawks. *Journal of Animal Ecology* 54: 241–253.
- Newton, I. 1994. The role of nest sites in limiting the numbers of hole-nesting birds: a review. *Biological Conservation* 70:265–276.
- Quinn, T. J. 1977. The effects of aggregation on line transect estimators of population abundance with applications to marine mammal populations. M.S. Thesis, University of Washington, Seattle, Washington, USA.
- Quinn, T. J. and R. B. Deriso. 1999. Quantitative fish dynamics. Oxford University Press, New York, New York, USA.
- Reed, W. J. 1979. Optimal escapement levels in stochastic and deterministic harvesting models. *Journal of Environmental Economics and Management* 6:350–363.
- Ricklefs, R. E. 1998. Evolutionary Theories of Aging: Confirmation of a Fundamental Prediction, with Implications for the Genetic Basis and Evolution of Life Span. *Am. Nat.* 152(1): 24-44.

Rodríguez, J. P., L. Fajardo, I. Herrera, A. Sánchez, and A. Reyes. *In press*. Conservation and management of the yellow-shouldered parrot (*Amazona barbadensis*) on the islands of Margarita and La Blanquilla, Venezuela. In: H. R. Akçakaya, M. Burgman, O. Kindvall, C. Wood, P. Sjoren-Gulve, J. Hattfield, and M. McCarthy, editors. *Species Conservation and Management: Case Studies*. Oxford University Press, Oxford, United Kingdom.

Rossi, M. N. and H. G. Fowler. 2004. Spatial and temporal population interactions between the parasitoids *Cotesia flavipes* and Tachinidae flies: considerations on the adverse effects of biological control practice. *Journal of Applied Entomology* 128(2): 112-119.

Rowley, I. and G. Chapman. 1991. The breeding biology, food, social organization, demography and conservation of the Major Mitchell or pink Cockatoo *Cacatua leadbeateri* on the margin of the Western Australian wheatbelt. *Australian Journal of Zoology* 39: 211-261.

Runge, M. C. and F. A. Johnson. 2002. The importance of functional form in optimal control solutions of problems in population dynamics. *Ecology* 83(5): 1357-1371.

Sæther, B.-E. and Ø. Bakke. 2000. Avian Life History Variation and Contribution of Demographic Traits to the Population Growth Rate. *Ecology* 81(3): 642-653.

Sandercock, B. K., S. R. Beissinger, S. H. Stoleson, R. R. Melland and C. R. Hughes. 2000. Survival Rates of a Neotropical Parrot: Implications for Latitudinal Comparisons of Avian Demography. *Ecology* 81(5): 1351-1370

Solberg, E. J., B.-E. Sæther, O. Strand, and A. Loison. 1999. Dynamics of a harvested moose population in a variable environment. *Journal of Animal Ecology* 68:186-204.

South, J. M. and T. F. Wright. 2002. Nestling sex ratios in the Yellow-naped Amazon: No evidence for adaptive modification. *Condor* 104(2): 437-440.

Southgate, D., J. Sanders and S. Ehui 1990. Resource Degradation in Africa and Latin America: Population Pressure, Policies, and Property Arrangements. *Journal of Agricultural Economics* pp. 1259-1263.

Stearns, S. C. 1992. *The evolution of life histories*. Oxford University Press, Oxford, UK.

Steelman, T. A. and J. Carmin 1998. Common Property, Collective Interests, and Community Opposition to Locally Unwanted Land Uses. *Society and Natural Resources* 11: 485-504.

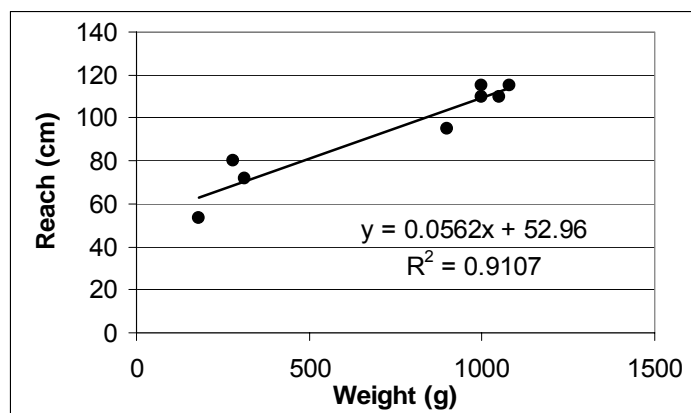
- Stoleson, S. H. and S. R. Biesenger. 1997. Hatching asynchrony in parrots: Boon or bane for sustainable use? Chap 7., pp. 157-180. IN: Behavioral approaches to conservation in the wild. Cambridge University Press. UK.
- Thomas, L., J. L. Laake, S. Strindberg, F. F. C. Marques, S. T. Buckland, D. L. Borchers, D. R. Anderson, K. P. Burnham, S. L. Hedley, J. H. Pollard and J. R. B. Bishop. 2003. Distance 4.1. Release "2". Research Unit for Wildlife Population Assessment, University of St. Andrews, UK. <http://www.ruwpa.st-and.ac.uk/distance/>
- Thompson, S. K. 1991. Adaptive cluster sampling: designs with primary and secondary units. *Biometrika* 47: 1103-1015.
- Thomsen, J. B. and A. Brautigam. 1991. Sustainable use of neotropical parrots. Pp. 359-379. IN: Neotropical wildlife use and conservation. (J. G. Robinson and K. H. Redford, Edits.). University of Chicago Press, Chicago, USA.
- Trent, S. 1999. Interpreting current levels of poaching of African elephants. *Oryx* 33(2): 94-95.
- Walters, C. 1981. Effects of Measurement Errors on the Assessment of Stock-Recruitment Relationships. *Canadian Journal of Fisheries and Aquatic Sciences*. 38: 704-711.
- Walters, C. 1986. *Adaptative Management of Renewable Resources*. MacMillan Publishing Company, New York. 374 pp.
- Walters, C. and A. M. Parma. 1996. Fixed exploitation rate strategies for coping with effects of climate change. *Canadian Journal of Fisheries and Aquatic Sciences*. 53:148-158.
- Williams, T. D., and S. Rodwell. 1992. Annual variation in return rate, mate and nest-site fidelity in breeding Gentoo and Macaroni Penguins. *Condor* 94: 636-645.
- Wilson, K. R. and D. R. Anderson 1985. Evaluation of a Density Estimator Based on a Trapping Web and Distance Sampling Theory. *Ecology* 66(4): 1185-1194.
- Wright, T. F., C. A. Toft, *et al.* 2001. Nest poaching in neotropical parrots. *Conservation Biology* 15(3): 710-720.

Appendix I

Average reach and weight data used for the regression to predict reach as a function of weight from eight species of Psittacidae. Source: Rodríguez, J. V. y J. I. Hernández. 2002. Loros de Colombia. Conservation International Colombia. 1ra. Edición. 478 p.

Genus	Species	Average reach (cm)	Average weight (g)
Aratinga	wagleri	53.5	180
Orthopsittaca	manilata	80	280
	severa	71.5	315
	militaris	95	900
Ara	ararauna	115	1000
	chloroptera	110	1000
	macao	110	1051
	ambigua	115	1080

The results of the linear regression of reach on weight are statistically significant ($t=7.821$, $df=6$, $p=0.00023067$).

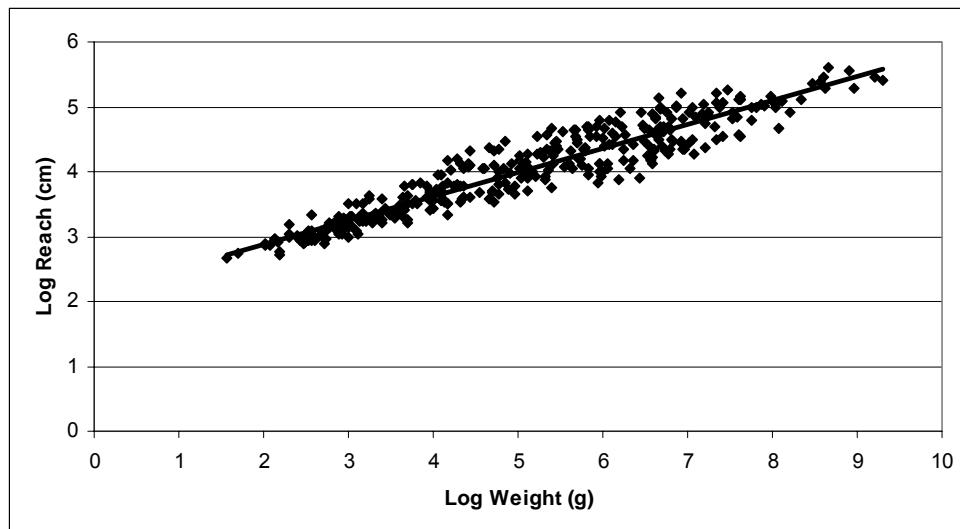


Appendix II

Double logarithm linear regression of average weight on average reach used based upon 363 species data. Source: <http://perso.wanadoo.fr/mousquey.piel/ornitho/encyclopedia/>

Regression Summary for Dependent Variable: LOG_REACH						
R= 0.94274547 R ² = 0.88876902 Adjusted R ² = 0.88846090						
F(1,361)=2884.5 p<0.0000 Std.Error of estimate: 0.23151						
N= 363		St. Err.		St. Err.		
	BETA	of BETA	B	of B	t(361)	p-level
Intercpt			2.149574	0.036643	58.66269	0
LOG_WEIGHT	0.942745	0.017553	0.367955	0.006851	53.70753	0

The breakpoint of the piecewise linear regression was estimated by the program (Statistica, Vers. 5.5).



Appendix III

Average reach and longevity data used for the regression to predict longevity as a function of reach from 27 species of different bird families. Sources are indicated in the table.

Common name	Genus	Species	Average reach (cm)	Source	Average Longevity (years)	Source
Common goose	Anser	domesticus	82.5	(1)	31	(2)
King penguin	Aptenodytes	patagonica	91	(3)	26	(3)
Grey heron	Ardea	Cinerea	110	(4)	24	(5)
Yellow-eyed penguin	Megadyptes	Antipodes	68	(6)	43.7*	(5)
Fulmar	Fulmaris	Glacialis	127	(4)	10	(5)
Street pigeon	Columba	Livia	66.5	(4)	35	(2)
Raven	Corvus	Corax	125	(4)	69	(2)
Griffon vulture	Gyps	Fulvus	260	(4)	41.42	(2)
Herring gull	Larus	Argentatus	153	(4)	44	(2)
European partridge	Perdix	Perdix	47	(4)	5	(2)
Ring-necked pheasant	Phasianus	Colchicus	80	(4)	27	(2)
Starling	Sturnus	vulgaris	37	(4)	15.83	(2)
Blue tit	Parus	caeruleus	20	(4)	9	(5)
European robin	Erithacus	rubecula	21	(4)	12	(5)
Redstart	Phoenicurus	phoenicurus	22	(4)	5.6*	(5)
Blackbird	Turdus	merula	36	(4)	7	(5)
Lapwing	Vanellus	vanellus	85	(4)	16	(5)
Alpine swift	Apus	melba	57	(4)	16	(5)
Common swift	Apus	apus	45	(4)	21	(5)
Sooty shearwater	Puffinus	griseus	101.5	(4)	27	(5)
Gannet	Sula	bassana	172.5	(4)	17	(5)
Whistling Swan	Cygnus	columbianus	195	(4)	18	(7)
Mew Gull	Larus	canus	120	(4)	18	(7)
Black-legged kittiwake	Rissa	tridactyla	107.5	(4)	12	(7)
Sparrowhawk	Accipiter	nisus	57.5	(4)	8	(7)
Pied flycatcher	Ficedula	hypoleuca	22	(4)	7	(7)
Great Tit	Parus	major	22	(4)	7	(7)

(1) http://212.187.155.84/pass_06june/Subdirectories_for_Search/SpeciesKingdoms/0Families_ACrAv_Anseriformes/anatidae/1ACrAvAn_anser/anser_anser

(2) Section 30, Life Spans: Animals. Page 106, in: Biology Data Book (P. H. Altman & D. S. Dittmer, Edit.) Federation of American Societies for Experimental Biology, Washington DC, USA. 1964.

(3) http://www.btinternet.com/~sa_sa/birdlife/kingpenguin.html

(4) <http://perso.wanadoo.fr/mousquey.piel/ornitho>

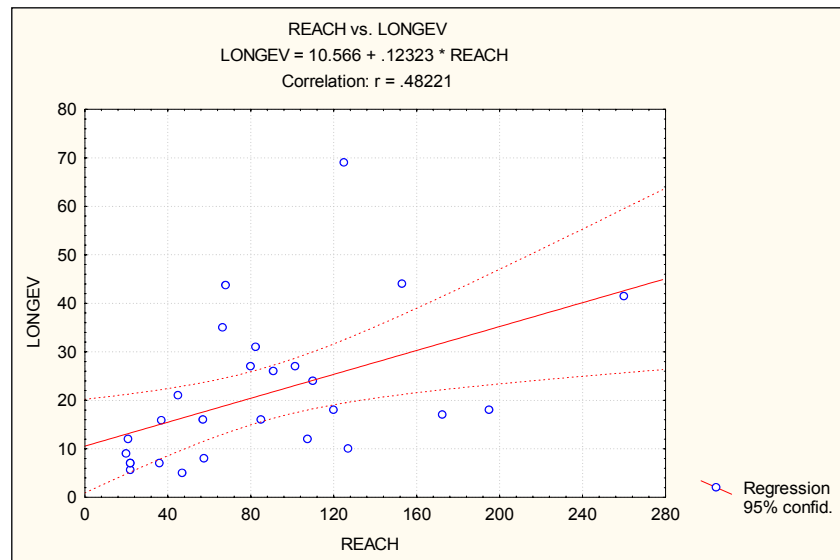
(5) Botkin & Miller (1982) Mortality rates and survival of birds

(6) <http://www.damisela.com/zoo/ave>

(7) Ricklefs, R. E. 1998. Evolutionary Theories of Ageing: confirmation of a Fundamental Prediction, with Implications for the Genetic Basis and Evolution of Life Spans. The American Naturalist 152(1): 24-44.

* Calculated on the basis of the average year mortality provided in the reference

The results of the linear regression of longevity on reach are statistically significant ($t=2.752$, $df=25$, $p=0.0108$)



Appendix IV

Reach estimates of the blue-fronted parrot based on the average weight estimated by simple and piecewise linear regression. The simple linear regression was based upon eight Psittacidae species (see Appendix I) and the double logarithm linear regression was based upon 361 bird species of a variety of families (see Appendices II).

Weight (g)	Predicted reach (cm)	
	Simple linear regression	Double log linear regression
400	75.44	77.82
500*	81.06	81.27
600	86.68	84.49

**This row in bold was outlined because it represents the most plausible weight value of the blue-fronted parrot*

Longevity estimates of the blue-fronted parrot based on the average reach estimated by simple and piecewise linear regressions. The bolded numbers represent the most reasonable average of weight value for the blue-fronted parrot (550 g).

Reach (cm) (estimated from eight Psittacidae species)	Estimated longevity (years)	Reach (cm) (estimated from 363 species of various families)	Estimated longevity (years)
75.44	19.85	77.82	20.27
81.06	20.55	81.27	20.61
86.68	21.24	84.49	20.92

Appendix V

Excerpts from several electronic mail correspondents with experience in *Amazona* rearing

Excerpts from:

Roy Berger

parrothouse@shaw.ca

Date: Wed, 26 May 2004 16:25:18 -0600

I have never found any scientific based information based on research that has been done on this subject. Everything I have read or discussed with others has been opinions.

Presumably you are talking about parrots kept in captivity. If you go to this link <http://www.google.com/search?q=amazon+parrot+life+span> you will find a large number of web sites each with different estimates on the life span of Amazon parrots. In addition, I have read a variety of text by well know authors and each author gives a different estimate.

I believe that a good estimate would be 30 to 50 years as an average life span. Naturally, like people, there are going to be some exception with some living longer.

There are many claiming that they know of birds with much longer life spans however the accuracy of these claims are often questionable. Because Amazon parrots have long life spans they often live out their lives with more than one owner and the true age is usually a guess and not fact.

I have a friend who has had an Amazon since he was a baby and is now 41 years old.

Excerpts from:

Rick Jordan

hatch111@earthlink.net

Date: Thu, 27 May 2004 12:30:23

Well, this is a subject that is very difficult to approach scientifically. Basically, a majority of the parrots with any "longevity" on them arrived here in the United States as imported birds from the wild. That means that no one really knows how old they were when they were trapped in the wild. Of course there were a number of Blue-fronted amazons that arrived in our quarantine system that were taken from the nest as nestlings, and still being hand-fed when they arrived in the stations back in the 1980s. Having said all that, to try and estimate how long amazons are living in captivity is still very difficult, but I will tell you of MY experience with this subject.

With regard to the Blue-fronted amazons, I'd have to estimate that "most" of the old imported birds are now dead. Of the thousands and thousands of this species that were imported in the 70s and 80s, there are only small groups that still exist in breeding collections, and a sporadic number that are still kept as pets. I have interviewed many people that have these birds (amazons in general) as pets, and it seems that some live as long as 25-28 years after importation before they succumb to liver diseases or other natural causes. People who keep hand-fed birds and actually know the ages of the birds are rare, but there are now some that can record ages into the late teens, ie: 15-19 years of age. RARELY and almost NEVER do I hear or speak with anyone that has a pet amazon that is over 30 years of age.

Based on all the people I know with pet or breeder amazons, I would have to say that the average lifespan in captivity is in the late teens- approximately 17-19, where the exceptional birds that live longer may live into their mid to late 20s.

Of course we have a few 'recorded' ages, very few that can be substantiated, claiming to have had an amazon for 40+ years. In my years in this business, I know of only two people that claim to have birds this old, One was a zoo in Houston, Texas, and the other was a private pet owner who inherited his double-yellow headed amazon from his father.

I hope this information, although not scientific, helps you to your goal. If I can be of further help, feel free to contact me.

**Excerpts from several Internet sites
with experience in *Amazona* rearing or trade**

**Amazon FAQ
Joanie Doss
amazons@ccountry.net**

***Amazona aestiva* (Blue Fronted Amazon)**

The amount of blue or yellow varies with the individual bird and subspecies. In the wild, they feed from the upper canopy and seldom explore the floor of the forest. They are intolerant of other birds during the breeding season. This is a very popular bird among bird fanciers. There were 97,000 Blue Fronts **legally** imported into the United States during a six year period. They are intelligent, beautiful, and most are excellent talkers. This bird has been a favorite pet bird in Europe for over a century. They seldom have a stand-off attitude and become deeply involved with humans. These birds are hardy and have long life spans. **The oldest known was 117 years.**

Cost:

Low price: \$400 - High price: \$1,000

Average price: \$655 ← **Prices as of 1997**

http://www.pethealthcare.net/html/body_choosing_an_amazon_parrot.html

Weight: average 10 - 20 ounces (300 - 600 grams)

Size: average 10 - 20 inches (25 - 45 cm) in length

Life span: 15 - 25 years (maximum 75 years)

Breeding: sexual maturity at 3 - 6 years

large environment is needed to breed this challenging bird

Brood Size: 2 - 7 eggs hatch in 17 -31 days, young leave the nest in 4 - 8 weeks

<http://www.seaworld.org/AnimalBytes/bluefrontedamazon.htm>

Genus and species: *Amazona aestiva*. Common Name: Blue-fronted Amazon

Fast Facts

Size: 37.5 cm (15 in.). **Weight:** 400 to 550 grams (14-19.25 oz)

Description: medium-sized parrot; primarily green; yellow face, with the blue feathers around beak; yellow and red patches on shoulder; dark gray feet and beak

Life span: up to 60-80 years

Sexual maturity: 3 to 5 years

Incubation: lasts about 28 days; clutches consist of 2 to 3 eggs; 60 to 75 days to fledge

Habitat: Southwestern Brazil, Paraguay, Bolivia , and northern Argentina; humid and dry forests, palm trees, and at the edge of rivers

Diet: seeds, fruits, nuts, berries, and leaves

Status: not listed by USFWS; CITES Appendix II

<http://www.parrotpassionsuk.com/thebirds/amason.htm>

There are 27 different Amazon species known in aviculture, it would be impossible to list every one! Here we will cover the more known species commonly kept as pet companions. **The average life span is 50 - 70 years.**

www.rasehallstudios.com/html/critters.html

Meet Peanut (Nutty), one of our kitties. Here's Carrie! Our Red Lord Amazon Parrot. We have had her since 1980 at 3 months old. **At 23 years old she looks pretty good** for being a old lady huh? (... that **in the wild, a parrot can live to be 85 years ...**)

<http://www.bbc.co.uk/nature/animals/pets/parrots.shtml>

Parrots in captivity can live for **40 to over 100 years (depending on the species).**

http://www.arndt-verlag.com/projekt/birds_3.cgi?Desc=E331.htm&Pic=331_1.JPG

This page is part of the LEXICON OF PARROTS. The complete Lexicon with 4 folders and display outers is available for just \$150 including shipping & handling

Amazona aestiva (Linné 1758)

2. Amazona a. xanthopteryx (Berlepsch 1896)

Yellow-winged Amazon - German: Gelbflügel-Blaustirnamazone

Description: as aestiva, but bend of wing yellow, variably interspersed with red and occasionally quite extensive. Length: 37 cm (14.5 ins)

Distribution: southwest Mato Grosso, Brazil, north and eastern Bolivia, Paraguay and northern Argentina.

Note: the classification of the species into two sub-species is not undisputed, as yellow on bend of wing taken as typical for xanthopteryx can occur in nominate form, even if not as consistently; aviculturists with Blue-fronted Amazons tend to differentiate incorrectly between sub-species by reference to extent of yellow-red colour on bend of wing.

Habitat: forest, palm groves, open savanna with woodland, cleared and cultivated areas to 1,600 m (4,800 ft).

Status: common; but endangered in some localities by extensive trapping

Habits: in pairs or small groups; seen during day either feeding or resting in tops of trees; then very quiet and only detected by falling food remains; not shy and can be approached; if alarmed, flies away screeching loudly; conspicuous and noisy during flight; pairs can be determined flying closely together; occasionally very large flocks gather towards evening in roosting trees; seasonal migrations; occasionally forages in plantations causing considerable damage; call raucous and loud.

Natural diet: fruits, berries, seeds, nuts, flowers and buds.

Breeding behaviour: breeding season October to March; nests in established trees; **generally one nest per hectare (2.5 acres) or about 100 nests/km²**; same nest hole used every year; egg measures 38,1 x 29,6 mm (1.50 x 1.16 ins).

Appendix VI

Summary of a Report from Eugeni Fernández Nogales, Gerona, Spain, on some survival and reproductive features of *Amazona aestiva* in captivity, extended with elements of a personal interview with Jorge Rabinovich on June 1, 2004 (EF= Eugeni Fernández). E-mail contact: info@isaugen.com, pb_malinois@hotmail.com

1. Mortality / survival / reproduction

- a. In the juvenile and adults individuals (after the first year of life), during the first 5 years mortality was zero. EF considers that, in general, when there is mortality it is usually for external causes: predation, illnesses, etc.
- b. Confronted with different survival curves (l_x) EF considers that the curve that best represents the survival in captivity would be located between the Type I and the Type I-II curves.
- c. From his own experience and from that from other aviaries EF considers that the maximum (extreme) longevity could be between the 50 and 60 years, while the reproductive life is in the order of 25-35 years.
- d. EF considers that reproductive life in the blue-fronted parrot in captivity begins around 4-5 years of age in the case of individuals born in captivity, and around 5-7 years of age in the case of individuals born under wild conditions and brought into the aviary. In the aviary EF obtained 100% of egg-laying by blue-fronted parrot pairs, but a success of births of about 80% (that is, at least around 80% of the pairs produced chicks).
- e. When the first batch of eggs is unsuccessful, if the failure takes place early in the reproductive period, (chicks are 5-10 or less days of age) in general after about 3 weeks the pair will usually lay a second batch of eggs.
- f. The reproductive schedule of the blue-fronted parrot is the following: (a) during years 1-5 they usually do not reproduce; (b) at the beginning (during the 6th and 7th year of age) egg laying takes place but is poor, either because they lay fewer eggs or because some of the eggs are infertile; (c) egg-laying is relatively stable until about 5-6 years before they cease to reproduce, that is, approximately the ages of 8-15 years, and (d) during those last 5-6 years of reproduction (that is to say, about years 25-30 of age) the egg-laying takes place but it is again quite poor (because they lay fewer eggs or because some of them are infertile).

2. Sex ratio

- a. EF has observed that sex ratios are extremely variable among different reproductive pairs. There are pairs that produce only male offspring, others that produce only female offspring, and others that produce a variable mix of both sexes.
- b. Based upon a limited statistics (about 15 pairs under reproduction for 5 years, that is, $N > 75$) the sex ratio was 30% female and 70% male.
- c. EF observed a conspicuous constancy in the sex ratio for specific pairs. For example, if a couple lays three eggs in stable form, and they leave two males and one female, the same pair will usually lay almost always three eggs in the following laying, and with the same sex ratio.

3. Behavior and general observations

- a. There is a high male/female fidelity. Once formed, pairs usually remain bonded during most of their life.
- b. There is a high nest. The same pairs usually occupy repeatedly the same nests.
- c. They are extremely perceptive and sensitive to "interferences" of different sorts: noises, leaks and flooding, presence or smell of predators, or nest alterations. In general, if a nest has to be abandoned because of some of those "interferences", frequently that nest will not be used the following year: they skip the following year's reproduction or they look for an alternative nest.
- d. When a single chick survives, while it remains very young (until about the 5th-10th day of age), the male parent tries to induce the female parent into laying a second batch. The female is reluctant to do so while the chick is in the nest. In some cases of very aggressive males, while they are trying to achieve a second egg laying from the female, they may end up killing the female in their efforts to copulate. In some other occasions they may kill the chick to induce the female to lay eggs again.

Appendix VII

Discussion of some recent findings from the literature on progeny sex-ratio in birds

About thirty years ago Trivers and Willard (1973) and Trivers (1974) proposed that offspring sex ratio should be associated with the quality of parental care likely to be provided to the offspring. It was hypothesized that parental “condition” could be central in influencing the sex ratio of offspring, “good condition” being associated with the conception of males. Although the emphasis on “condition” has been questioned (Grant, 2003) and a shift of the focus from good condition to the dominance-testosterone link is taking place, its importance on sex ratio determination is still a strong and evolutionarily powerful concept. Some of the basic consequences of the hypothesis is that parents are expected to overproduce the less costly sex under poor food conditions. The concept behind it is that the ability to alter primary sex ratios has the potential to increase a breeding individual's fitness.

However, the evidences for the application of this hypothesis in birds are still quite contradictory. For example, in the Ural owl, *Strix uralensis*, Brommer *et al.* (2003) studied the owls' parental feeding investment and sex allocation during poor-quality years, and showed that the owls produced significantly more males (56%). However the food delivered to the broods without chick mortality (N=83) showed no clear sex-specific investment, and nestling mortality was equal in both sexes. Thus, evidence for an investment-driven sex allocation is weak. Additionally, neither laying date, brood size nor the female's condition correlated with offspring sex ratios.

On the other hand Ewen *et al.* (2003) studying the facultative control of offspring sex in the cooperatively breeding bell miner, *Manorina melanophrys* tried to determine whether female breeders adjusted the sex ratio of broods to produce more of the philopatric sex when food resources were high and more of the dispersing sex when food resources were low. These authors found that there was a clear evidence for such facultative control of sex ratio by female bell miners.

Another prediction of this hypothesis is that mothers are supposed to overproduce male or female eggs when the relative fitness gains from one sex are higher and outweigh the costs of manipulation. But this sex-ratio changes appear to be subtle. Using a molecular technique to identify the sex of early embryos to distinguish from chick mortality, Arnold *et al.* (2003) studied the effect of maternal nutrition on zebra finch (*Taeniopygia guttata*) egg sex ratios after 2 days of incubation, and found no overall bias in the sex ratio of eggs laid and sex did not differ with relative laying order under any diet regime. However, mothers on a low-quality diet did produce a female bias in small clutches and a slight male bias in large clutches. On a high-quality diet, mothers produced a male bias in small clutches and a female bias in large clutches. Those on a standard diet produced a roughly even sex ratio, irrespective of clutch size. These observed biases in egg sex are partly in line with predictions that, in this species, daughters suffer disproportionately from poor rearing conditions. Thus, when relatively malnourished, mothers should only rear daughters in small broods and vice versa.

Andersson *et al.* (2003) decided to test seasonal change in sex ratio as a possibility for predicting adaptive adjustment in offspring sex ratio by females. The test was carried on in two sister species, the Common sandpiper and the Spotted sandpiper *Actitis hypoleucos* and *A. macularia*. In the monogamous Common sandpiper, where males are the most competitive sex, these authors found a change from mainly sons in early clutches to mainly daughters in late clutches. They confirmed that this seasonal adjustment of clutch sex ratio took place within the female before the eggs were laid, not by differential egg or chick survival. On the other hand in the Spotted sandpiper, which is polyandrous with partly reversed sex roles, there was no seasonal trend from sons to daughters in this species.

The sex ratio determination by parents becomes more complex because it can also be affected by the order of mating. Albrecht and Johnson (2002) tested the possibility of a manipulation of offspring sex ratio by second-mated female house wrens (*Troglodytes aedon*), and found that second-mated females typically receive little or no male parental assistance and fledge fewer and lower-quality young compared with first-mated females. They also found that mean fledgling sex ratios in nests of second-mated females were more female-biased than fledgling sex ratios in nests of first-mated females. However they found no evidence of either sex-biased nestling mortality or sex-biased parental provisioning, suggesting that females are responding to their status as second-mated females and to the associated low-quality parental care that their young are likely to receive by producing female-biased clutches rather than manipulating the offspring sex ratio through sex-biased nestling mortality.

Sandercock *et al.* (2000) found that, overall, in the Green-rumped Parrotlet (*Forpus passerinus*) males outnumbered females 1.7 : 1.0. Although they found that early nest loss affected female survival, presumably because females are vulnerable while they incubate and brood the young alone, they concluded that small differences in adult mortality are unlikely to explain the male-biased sex ratio in adult parrotlets, whereas differential mortality and differential emigration between female and male fledglings appear to be important.

Apart from the study of the Green-rumped Parrotlet (Sandercock *et al.*, 2000), and the study of the progeny sex ratio of *A. aestiva*, the progeny sex ratio of only two other species of parrots have been studied: in the Yellow-naped Amazon, *Amazona auropalliata* (South and Wright, 2002) and in the Kakapo, *Strigops habroptilus* a large, flightless, nocturnal parrot, endemic to New Zealand.

In the case of the blue-fronted parrot a limited experience with field *A. aestiva* populations (Fernandes Seixas and Miranda Mourão, 2002) found that the sex ratio for a sample of 27 chicks from 15 nests during the reproductive season was not differing significantly from 1:1.

South and Wright (2002), working with the Yellow-naped Amazon (*A. auropalliata*) determined that overall, the sex ratio of nestlings was 51% male (39 out of 77), which did

not represent a significant departure from unity; they also found that the mean proportion of males per brood ($n = 37$) was 49%, which did not differ from unity, nor did the sex ratio of first-born or second-born nestlings.

The only exception in the limited experience with the progeny sex ratios in the parrot family is the kakapo, *Strigops habroptilus*, a species with a polygynous “lek” mating system. This species is critically endangered, with a world population of ca. 62 individuals and a male-biased adult sex ratio, with adult males typically weigh 30–40% more than females (Clout *et al.*, 2002). The kakapo has a skewed sex ratio towards males (Trewick, 1996), however, as it is subject to intensive conservation efforts, including the provision of supplementary food to wild birds to encourage successful nesting, this biased sex ratio seems to be associated to this artificial feeding condition. Clout *et al.* (2002) used data on the sex ratio of progeny of female kakapo that had or had not received supplementary food, to test the hypothesis that supplementary feeding might cause a male-biased offspring sex ratio. Their results show that there was a significant excess of males in the clutches of females provided with supplementary food.

Under this light I decided to be cautious with the information that, in *A. aestiva*, under extremely favorable feeding and rearing conditions, progeny sex ratio apparently is strongly male-biased (up to 30 males:70 females ratios; Eugeni Fernández Nogales, *pers. comm.*; see Appendix VI). The careful experience with the kakapo, and the two hard data case studies with field populations of *A. auropalliata* and *A. aestiva* (despite their low sampling numbers) strongly suggest that we should not expect a marked alteration in the 1:1 offspring sex ratio in the blue-fronted parrot.

As the progeny sex ratio is a critical value in the application of Lotka’s equation for estimating r , because all calculations are carried out as if the whole population would be a “virtual” exclusively female population, with the assumption that there will always be enough males available as to fertilize all females, it was important to justify the use of a 1:1 progeny sex ratio (i.e., 50% males and 50% females) in the blue-fronted parrot despite very little information is available, either under wild or commercial rearing conditions.

Literature cited in Appendix VII

Albrecht, D. J. and L. S. Johnson. 2002. Manipulation of offspring sex ratio by second-mated female house wrens. *Proceedings of the Royal Society Biological Sciences Series B* 269(1490): 461-465.

Andersson, M., J. Wallander, L. Oring, E. Akst, J. M. Reed and R. C. Fleischer. 2003. Adaptive seasonal trend in brood sex ratio: Test in two sister species with contrasting breeding systems. *Journal of Evolutionary Biology* 16(3): 510-515.

- Arnold, K. E., R. Griffiths, D. J. Stevens, K. J. Orr, A. Adam and D. C. Houston. 2003. Subtle manipulation of egg sex ratio in birds. *Proceedings of the Royal Society Biological Sciences Series B* 270(Supplement 2): S216-S219.
- Brommer, J. E., P. Karell, T. Pihlaja, J. N. Painter, C. R. Primmer and H. Pietiäinen. 2003. Ural owl sex allocation and parental investment under poor food conditions. *Oecologia (Berlin)* 137(1): 140-147.
- Clout, M. N., G. P. Elliott and B. C. Robertson. 2002. Effects of supplementary feeding on the offspring sex ratio of kakapo: A dilemma for the conservation of a polygynous parrot. *Biological Conservation* 107(1): 13-18.
- Ewen, J. G., R. H. Crozier, P. Cassey, T. Ward-Smith, J. N. Painter, R. J. Robertson, D. A. Jones and M. F. Clarke. 2003. Facultative control of offspring sex in the cooperatively breeding bell miner, *Manorina melanophrys*. *Behavioral Ecology* 14(2): 157-164.
- Fernandes Seixas, G. H. and G. de Miranda Mourão. 2002. Nesting success and hatching survival of the Blue-fronted Amazon (*Amazona aestiva*) in the Pantanal of Mato Grosso do Sul, Brazil. *J. Field Ornithol.* 73(4): 399-409.
- Grant, V. J. 2003. The maternal dominance hypothesis: questioning Trivers and Willard. *Evolutionary Psychology* 1: 96-107.
- Sandercock, B. K., S. R. Beissinger, S. H. Stoleson, R. R. Melland and C. R. Hughes. 2000. Survival Rates of a Neotropical Parrot: Implications for Latitudinal Comparisons of Avian Demography. *Ecology* 81(5): 1351-1370
- South, J. M. and T. F. Wright. 2002. Nestling sex ratios in the Yellow-naped Amazon: No evidence for adaptive modification. *Condor* 104(2): 437-440.
- Trewick, S. A. 1996. On the skewed sex ratio of the Kakapo *Strigops habroptilus*: sexual and natural selection in opposition? *Ibis* 139:652-663.
- Trivers, R. L. 1974. Parent-Offspring Conflict. *American Zoologist* 14(1): 249-264.
- Trivers, R. L. and D. E. Willard. 1973. Natural selection of parental ability to vary the sex ratio of offspring. *Science* 179: 90-92.

Appendix VIII

FORTTRAN program developed to simulate the growth of a blue-fronted parrot population, assigning to it different values of its two parameters, r and K , and including the “fixed escapement” rule of population management, to select –for each combination r and K – the escapement level that maximizes annual offtake.

```

C
C... Modelo de manejo sustentable del loro hablador (Amazona aestiva)
C
C... Jorge Rabinovich - abril - julio de 2004
C
C+++++
C+      Modelo logístico variando el valor de r y de Kmedia      +
C+++++
C
C... Se utiliza un modelo de crecimiento poblacional logístico,
C... y con cosecha tipo escapes fijos ("Bang-Bang")
C
C... Ecuación de crecimiento poblacional:
C      -----
C...  $P(i)=P(i-1)*(K(i)*Rmax)/(K(i)+(P(i-1)*Rmax-1))$ 
C
C... donde:
C... Rmax= Tasa finita de crecimiento poblacional máxima= exp(r)
C      (y donde r es la tasa instantánea de crecimiento poblacional
C      per cápita)(esto supone que no hay competencia intraespecífica)
C... P(i)= Tamaño poblacional de loros en el año i
C... K(i)= Capacidad de carga de loros en el año i
C
C... La Capacidad de Carga de cada año i (es decir, K(i)) se determina
C... asignando un valor medio al parámetro K (Kmedia) y un coeficiente
C... de variación con el cual se extrae un valor aleatorio (CVK) de K
C... de acuerdo a una distribución normal
C
C... La regla de manejo Bang-Bang se puede expresar como:
C... Si en el momento t, la densidad poblacional ( $D_t$ ) es igual o menor c... que un cierto
C... valor denominado densidad de escape o densidad
C... umbral ( $D_U$ ), entonces la extracción (Z) es cero; si la densidad c... poblacional es
C... mayor que la densidad de escape ( $D_U$ ), entonces la
C... extracción (Z) es toda aquella densidad de loros popr encima de
C... de la densidad de escape ( $D_U$ )
C
C
C      DIMENSION P(100),PP(100),RSENSI(10),RESUVEDA(10,100),
1 RESUCOSE(10,100),RESUEXTR(10,100)
C      DIMENSION RESUUFIN(10),RESUCMAX(10),RESUCACU(10),COSEPROV(100)
C      DIMENSION RESUMATE(10,10),DENSESCA(10,10),IND_K(10),IND_R(10),
1 RESUPEST(10,10)
C      REAL K(100),Kmedia,Ksensi(10)
C
C... RESUVEDA(i,j) y RESUCOSE(i,j) son variables donde se guardan los
C... resultados de sensibilidad de las simulaciones sin y con cosecha
C... respectivamente, donde i lazo de sensibilidad y j años de
C... simulación
C
C      OPEN(2,FILE='LOGISENSI.SAL',status='unknown')
C      OPEN(3,FILE='MATRIZSEN.SAL',status='unknown')
C
C      WRITE(*,*) 'Desea K aleatorio? (0= NO, 1= SI)'
C      READ(*,*) IKAL
C
C      SUPERF=20000.
C      IRPAT=1111 ! Semilla de la rutina de general números al azar
C      N=30       ! No. de años a simular
C      POBIN=1    ! Densidad inicial de loros (loros/km2)
C      Kmedia= 45 ! Capacidad de carga media de (loros/km2)
C      CVK=0      ! Coeficiente de variación porcentual

```

```

R=0.6          ! R es la tasa instantánea de crecimiento natural
IF(IKAL.EQ.0) GOTO 10
WRITE(*,*)' Coeficiente de variación de K (OJO: EN PORCENTAJE)'
READ(*,*)CVK
10  CONTINUE
c
c... Se inicia el lazo de sensibilidad para probar los efectos de
c... diversos valores de K
c
      KONT_K=0
      DO 600 IK=2,14,2
      KONT_K=KONT_K+1
      IND_K(KONT_K)=KONT_K
      Ksensi(IND_K(KONT_K))=IK
      Kmedia=Ksensi(IND_K(KONT_K))
c
c... Se inicia el lazo de sensibilidad para probar los efectos
c... de diversos valores de r
c
      KONT_R=0
      DO 500 IR=1,6
      KONT_R=KONT_R+1
      IND_R(KONT_R)=KONT_R
      RSENSI(IR)=EXP(FLOAT(IR)/10.)
      R=RSENSI(IR)
      RESUVEDA(IR,1)=POBIN
      RESUCOSE(IR,1)=POBIN
      RESUEXTR(IR,1)=0
c
c... Se realiza un lazo de simulación sin cosecha
c
      PP(1)=POBIN
      DO 50 I=2,N
      K(I)=Kmedia
      IF(IKAL.EQ.0)GOTO 45
      CALL AZAR(IRPAT,AZ,Kmedia,CVK)
      K(I)=AZ
45    CONTINUE
      IF(PP(I-1).LT.K(I)) GOTO 47
      PP(I)=K(I)
      GOTO 48
47    CONTINUE
      PP(I)=PP(I-1)*((K(I)*R)/(K(I)+(PP(I-1)*(R-1))))
      IF(PP(I).GT.K(I))PP(I)=K(I)
48    CONTINUE
      RESUVEDA(IR,I)=PP(I)
50    CONTINUE
c
c... Se inicia el lazo de simulación para el valor de Bang-Bang
c
      COSMAX=0
      DO 200 J=1,100
      COSACU=0
      UM=FLOAT(J)
c
c... Se realiza el lazo de simulación con cosecha de la población
c
      P(1)=POBIN
      IRPAT=1111
      DO 100 I=2,N
      K(I)=Kmedia
      IF(IKAL.EQ.0)GOTO 55
      CALL AZAR(IRPAT,AZ,Kmedia,CVK)
      K(I)=AZ
55    CONTINUE
      IF(K(I).EQ.0) STOP ' OJO K(I)= 0 '
      IF(P(I).GT.K(I))P(I)=K(I)
      P(I)=P(I-1)*((K(I)*R)/(K(I)+(P(I-1)*(R-1))))
      IF(P(I).GT.K(I))P(I)=K(I)
      PROPCOS=0
      IF(P(I).GT.UM)PROPCOS=(P(I)-UM)/P(I)

```

```

        COSECHA=P(I)*PROPCOS
        COSACU=COSACU+COSECHA
        COSEPROV(I)=COSECHA
        P(I)= P(I) - COSECHA
100    CONTINUE
150    FORMAT(I3,';',5(F10.2,';'))
C
C... SELECCIONO EL UM y OTRAS VARIABLES SI SE MAXIMIZA LA
c... COSECHA ACUMULADA
C
        IF(COSACU.LE.COSMAX) GOTO 175
        COSMAX=COSACU
        RESUCMAX(IR)=COSMAX
        COSEFIN=COSACU
        RESUCACU(IR)=COSEFIN
        UMFIN=UM
        RESUUFIN(IR)=UMFIN
        DENSESCA(IND_R(KONT_R),IND_K(KONT_K))=UMFIN
C... EN P015_30 GUARDO EL PROMEDIO DE LA EXTRACCIÓN DE
c... LOS AÑOS 15 A 30 DE SIMULACIÓN
        P015_30=0
        P0ESTAB=0
        DO 165 IRES= 2,N
        IF(IRES.GE.15)P015_30=P015_30+COSEPROV(IRES)
        IF(IRES.GE.15)P0ESTAB=P0ESTAB+P(IRES)
        RESUCOSE(IR,IRES)=P(IRES)
        RESUEXTR(IR,IRES)=COSEPROV(IRES)
165    CONTINUE
        IF((N-15.+1.).eq.0) stop '(N-15.+1.) es cero'
        P015_30=P015_30/(N-15.+1.)
        P0ESTAB=P0ESTAB/(N-15.+1.)
C... RESUMATE(IR,IK) ES LA MATRIZ DONDE SE GUARDA LA EXTRACCIÓN PROMEDIO
C... ESTABILIZADA DE LOS ÚLTIMOS 15 AÑOS (DE LOS 30 AÑOS DE SIMULACIÓN)
        RESUMATE(IND_R(KONT_R),IND_K(KONT_K))=P015_30
        RESUPEST(IND_R(KONT_R),IND_K(KONT_K))=P0ESTAB
175    CONTINUE
C
200    CONTINUE
c
C... Escribo los encabezados
c
        WRITE(2,275)
275    FORMAT('Año',';', 'K',';', 'Loros sin cosechar',';',
1      'Loros bajo cosecha',';', 'Cosecha',';', 'UMFIN')
c
c... Escribo los resultados para cada valor de r
c
        write(2,*) ' r= ',RSENSI(IR),' K(I)= ',Ksensi(IND_K(KONT_K)), ' CVK= ',CVK
        DO 300 I=1,N
        WRITE(2,150)I,K(I),RESUVEDA(IR,I),RESUCOSE(IR,I),
1      RESUEXTR(IR,I),RESUUFIN(IR)
300    CONTINUE
500    CONTINUE
600    CONTINUE
C
C... PREPARO LA SALIDA DE LA MATRIZ DE LOS PROMEDIOS
c... DE EXTRACCIÓN DE LOS ÚLTIMOS 15 AÑOS EN FUNCION
c... DE LOS DISTINTOS VALORES DE R Y K UTILIZADOS
C
        WRITE(3,*) ' KKR= ',KKR
        WRITE(3,650)CVK,SUPERF
650    FORMAT(' Variación de K= ',F6.0,'% - Superficie= ',F6.0,' KM2')
        WRITE(3,725)(ALOG(RSENSI(IR)),IR= 1,KONT_R)
        DO 700 IK=1,KONT_K
        WRITE(3,750)Ksensi(IK),((RESUMATE(IR,IK)*SUPERF),IR=1,KONT_R)
700    CONTINUE
725    FORMAT('K',';',R= ',10(F6.2,';'))
750    FORMAT(5(F5.1,10(';',F10.1)))
c
C... PREPARO LA SALIDA DE LA MATRIZ DE LAS DENSIDADES DE ESCAPE
C

```



```

      WRITE(3,875)
      WRITE(3,*)' KKR= ',KKR
      WRITE(3,850)CVK,SUPERF
850   FORMAT(' Variación de K= ',F6.0,'% - Superficie= ',F6.0,' KM2')
      WRITE(3,825)(ALOG(RSENSI(IR)),IR= 1,KONT_R)
      DO 800 IK=1,KONT_K
      WRITE(3,860)Ksensi(IK),((DENSESCA(IR,IK)),IR=1,KONT_R)
800   CONTINUE
825   FORMAT('K;',';R= ',10(F6.2,','))
860   FORMAT(5(F5.1,10(',';','F10.1)))
875   FORMAT('// Valores de la densidad de escape')
C
C
C... PREPARO LA SALIDA DE LA MATRIZ DE LAS DENSIDADES
C... POBLACIONALES ESTABLES CON COSECHA
C
      WRITE(3,975)
      WRITE(3,*)' KKR= ',KKR
      WRITE(3,850)CVK,SUPERF
      WRITE(3,825)(ALOG(RSENSI(IR)),IR= 1,KONT_R)
      DO 900 IK=1,KONT_K
      WRITE(3,860)Ksensi(IK),((RESUPEST(IR,IK)),IR=1,KONT_R)
900   CONTINUE
975   FORMAT('// Valores de la densidad estabilizada')
C
      CLOSE(2)
      CLOSE(3)
      STOP 'Fin normal de Logistica con sensibilidad a R y K'
      END
C
c... Extracción de un valor de una variable al azar
c... con distribucion normal
c... Uso el metodo de The Ecological Detective (Hilborn & Mangel)
C
      SUBROUTINE AZAR(IRPAT,AZ,Kmedia,CVK)
      REAL Kmedia
      promedio=Kmedia
      desvesta=CVK/100*Kmedia
      VALORPI=3.1415926536
      T0=RANDS(IRPAT)
      U1=RANDS(IRPAT)
      U2=RANDS(IRPAT)
      Z1= SQRT(-2*LOG10(U1))*COS(2*VALORPI*U2)
      Z2= SQRT(-2*LOG10(U1))*SIN(2*VALORPI*U2)
      DIAPRIM1=promedio+desvesta*Z1
      DIAPRIM2=promedio+desvesta*Z2
      AZ=DIAPRIM1
100  continue
      RETURN
      end
C
      FUNCTION RANDS(IX)
      -----
C   NOTA: Este programa fue tomado de los programas estadisticos BIOM
C   de Rholf (complementario del libro Biometry). Para mas detalles
C   ver el programa RANDTST del paquete BIOM que es el que lo usa
C
C   -RANDS-      PORTABLE RANDOM NUMBER GENERATOR
C                 USING THE RECURSION
C                 IX = IX*A MOD P
C
C   FROM:  SCHRAGE, L. 1979. A MORE PORTABLE FORTRAN RANDOM NUMBER
C           GENERATOR.  ACM TRANS. ON MATH. SOFTWARE, 5:132-138
C   ON INPUT IX MUST BE SET TO AN INTEGER 0<IX<2147483647
C   RETURNS REAL NUMBER 0.<RANDS<1. AND IX IS UPDATED
C - - - - -
      INTEGER A, P, IX, B15, B16, XHI, XALO, LEFTLO, FHI, K
C
C   7**5, 2**15, 2**16, 2**31-1
      DATA A/16807/,B15/32768/,B16/65536/,P/2147483647/
C - - - - -

```

```

C
C   GET 15 HI ORDER BITS OF IX
C   XHI = IX/B16
C   GET 16 LO BITS OF IX AND FORM LO PRODUCT
C   XALO=(IX-XHI*B16)*A
C   GET 15 HI ORDER BITS OF LO PRODUCT
C   LEFTLO = XALO/B16
C   FORM THE 31 HIGHEST BITS OF FULL PRODUCT
C   FHI = XHI*A + LEFTLO
C   GET OVERFLO PAST 31ST BIT OF FULL PRODUCT
C   K = FHI/B15
C   ASSEMBLE ALL THE PARTS AND PRESUBTRACT P
C   THE PARENTHESES ARE ESSENTIAL
C   IX = (((XALO-LEFTLO*B16) - P) + (FHI-K*B15)*B16) + K
C   ADD P BACK IN IF NECESSARY
C   IF (IX .LT. 0) IX = IX + P
C   MULTIPLY BY 1/(2**31-1)
C   RANDS = FLOAT(IX)*4.656612875E-10
C   RETURN
C   END

```

Appendix IX

Delimitation of the sampling area for the application of the nearest neighbor and point-to-nearest object methods of density estimation.

1. Basic concept

- a. In a certain area there are n trees and a GPS mapping system of them is available. The procedure starts drawing “huge” circles (see point 2 below), one for each tree. While all circles are *united* (see point 1b below) the radius of all circles is reduced in the same magnitude (for the criterion to do so see point 2c below). When the first separation (see point 1c below) is detected the process of reducing the radii is ended.
- b. United circles are those circles that comply the following condition: two randomly selected circles A and B that do not intersect each other must satisfy the following condition: there must exist C_1, C_2, \dots, C_n such that A intersects with C_1 , C_1 with C_2 , ..., C_{n-1} with C_n , and C_n with B.
- c. Separation occurs if condition (b) above is not satisfied.

2. Details of the algorithm used

- a. There are n trees, and each tree has its XY Cartesian position determined selecting as point of origin a point such that all trees fall in the first quadrant.
- b. A circle is assigned to each tree. The center of a circle will be the Cartesian position (XY) of a tree (see point 1). The initial radius R of each circle must be sufficiently large as to enclose all the other trees in the area.
- c. The radii R of all circles are reduced simultaneously in the same small and arbitrary magnitude of one meter.
- d. For each circle C_i the distance D to all other circles is measured.
- e. If in point 4 a certain C_j is found such that $D_{ij} > 2R$, then circles i y j are considered “separated”. If no C_1, C_2, \dots, C_n exist such that C_i intersects with C_1 , C_1 with C_2 , ..., C_{n-1} with C_n , and C_n with C_j then the process of area delimitation ends.
- f. Steps 2c-2e are repeated until the first separation is found.

Appendix X

C++ program developed to calculate the delimitation of the surface area where the random points could land to estimate fledgling density using the the nearest neighbor and point-to-nearest object methods. It also selects a certain number of random points to make a quadrat sampling.

Program Uloros.cpp

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "ULoros.h"
#include <RutinasUtilesCPP.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "UTablasEstadisticas.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

const double pi = 3.1415926535897932384626433832795;
Perinola *perinola = new Perinola();

TMsgDlgButtons botonesSiNo;

Punto::Punto(){
}

Punto::Punto(long double _x, long double _y){
    x = _x;
    y = _y;
}

Punto::Punto(Punto *p){
    x = p->x;
    y = p->y;
}

double Punto::distanciaA(Punto *p){
    return sqrt(pow(x - p->x, 2) + pow(y - p->y, 2));
}

TablaDeFrecuencia::TablaDeFrecuencia(): TObject(){
    limite = 50;
    valores = (int *)malloc(sizeof(int) * limite);
    this->vaciar();
}

void TablaDeFrecuencia::agregar(int cantidad){
    if(valores[cantidad] == -1)
        valores[cantidad] = 1;
    else{
        valores[cantidad]++;
    }
}

void TablaDeFrecuencia::vaciar(){
    for(int x = 0; x < limite; x++)
        valores[x] = -1;
}

int TablaDeFrecuencia::cantidadMaxima(){
    int x = limite - 1;
```

```

    while((x >= 0) && (valores[x] == -1)){
        x--;
    }
    return x;
}

int TablaDeFrecuencia::frecuenciaDe(int cantidad){
    if(valores[cantidad] == -1)
        return 0;
    else
        return valores[cantidad];
}

Cuadrado::Cuadrado() : Figura(){
    punto1 = new Punto();
    punto2 = new Punto();
    punto3 = new Punto();
    punto4 = new Punto();
    miCentro = new Punto();
}

__fastcall Cuadrado::~Cuadrado(){
    delete punto1;
    delete punto2;
    delete punto3;
    delete punto4;
    delete miCentro;
}

Cuadrado::Cuadrado(Punto *_punto1, Punto *_punto2, Punto *_punto3, Punto
*_punto4):Figura(){
    punto1 = new Punto(_punto1);
    punto2 = new Punto(_punto2);
    punto3 = new Punto(_punto3);
    punto4 = new Punto(_punto4);

    miCentro = new Punto();
    miCentro->x = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)) +
        (max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)) -
        min(min(punto1->x, punto2->x), min(punto3->x, punto4->x))) / 2;
    miCentro->y = min(min(punto1->y, punto2->y), min(punto3->y, punto4->y)) +
        (max(max(punto1->y, punto2->y), max(punto3->y, punto4->y)) -
        min(min(punto1->y, punto2->y), min(punto3->y, punto4->y))) / 2;

    //calculo las coordenadas del cuadrado (rotandolo) tal que los bordes
    //queden paralelos al eje x e y
    long double angulo, altura;
    altura = sqrtl(powl(punto1->x - punto2->x, 2) + powl(punto1->y - punto2->y, 2));
    if(punto1->x < punto2->x){
        angulo =(punto2->x - punto1->x) / altura;
    }
    else
        angulo =(punto1->x - punto2->x) / altura;

    //angulo en realidad es el coseno del angulo que debo rotar
    if(punto1->y < punto2->y){
        xminrotado = punto1->x;
        ymaxrotado = punto1->y;
        xmaxrotado = punto1->x + altura;
        yminrotado = punto1->y - altura;
    }

    rotacion = angulo;
}

void Cuadrado::armarEn(TPoint punto, long double lado, long double rotacion){
    if(rotacion != 0)
        ShowMessage("Error: falta terminar el metodo Cuadrado::armarEn");

    punto1->x = punto.x;
    punto1->y = punto.y;

```

```

    punto2->x = punto.x + lado;
    punto2->y = punto.y;
    punto3->x = punto.x;
    punto3->y = punto.y - lado;
    punto4->x = punto.x + lado;
    punto4->y = punto.y - lado;
}

void Cuadrado::setPuntos(Punto *_punto1, Punto *_punto2, Punto *_punto3, Punto *_punto4){
    punto1->x = _punto1->x;
    punto1->y = _punto1->y;
    punto2->x = _punto2->x;
    punto2->y = _punto2->y;
    punto3->x = _punto3->x;
    punto3->y = _punto3->y;
    punto4->x = _punto4->x;
    punto4->y = _punto4->y;

    miCentro->x = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)) +
        (max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)) -
        min(min(punto1->x, punto2->x), min(punto3->x, punto4->x))) / 2;
    miCentro->y = min(min(punto1->y, punto2->y), min(punto3->y, punto4->y)) +
        (max(max(punto1->y, punto2->y), max(punto3->y, punto4->y)) -
        min(min(punto1->y, punto2->y), min(punto3->y, punto4->y))) / 2;

    //calculo las coordenadas del cuadrado (rotandolo) tal que los bordes
    //queden paralelos al eje x e y
    long double angulo, altura;
    altura = sqrtl(powl(punto1->x - punto2->x, 2) + powl(punto1->y - punto2->y, 2));
    if(punto1->x < punto2->x){
        angulo =(punto2->x - punto1->x) / altura;
    }
    else
        angulo =(punto1->x - punto2->x) / altura;

    //angulo en realidad es el coseno del angulo que debo rotar
    if(punto1->y < punto2->y){
        xminrotado = punto1->x;
        ymaxrotado = punto1->y;
        xmaxrotado = punto1->x + altura;
        yminrotado = punto1->y - altura;
    }

    rotacion = angulo;
}

bool Cuadrado::intersectasCon(Figura *c){
    return this->existeInterseccion((Cuadrado*)c) || ((Cuadrado*)c)-
>existeInterseccion(this);
}

bool Cuadrado::existeInterseccion(Cuadrado *c){
    bool ok = c->tePerteneceElPunto(punto1);
    ok = ok || c->tePerteneceElPunto(punto2);
    ok = ok || c->tePerteneceElPunto(punto3);
    ok = ok || c->tePerteneceElPunto(punto4);
    return ok || c->tePerteneceElPunto(this->centro());
}

bool Cuadrado::tePerteneceElPunto(Punto *p){
    long double test, divisor;
    bool ok = true;
    Punto *a, *b, *c;

    //ab^2 = bc^2 + ac^2 - 2 cos(c) bc ac
    //despejando cos(c)
    //cos(c) = (bc^2 + ac^2 - ab^2) / (2 bc ac)

    //el punto p estará dentro del cuadrado si cumple lo siguiente
    //para sus cuatro lados:
    //para el lado que une los puntos f y g,

```

```

//el angulo que forma las recta (f, p) con (f, g)
//es menor o igual que 90 grados, y
//el angulo que forma las recta (g, p) con (f, g)
//es menor o igual que 90 grados.

//veo el angulo entre el punto 1 y el 2
try{
    //lado ab = punto1, p
    a = punto1; b = p; c = punto2;
    divisor = b->distanciaA(c) * a->distanciaA(c) * 2;
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }

    //lado ab = punto2, p
    a = punto2; b = p; c = punto1;
    divisor = b->distanciaA(c) * a->distanciaA(c) * 2;
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }
}
catch (Exception &e){
    test = 0;
};

//veo el angulo entre el punto 2 y el 4
try{
    //lado ab = punto2, p
    a = punto2; b = p; c = punto4;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }

    //lado ab = punto4, p
    a = punto4; b = p; c = punto2;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }
}
catch (Exception &e){
    test = 0;};

```

```

//veo el angulo entre el punto 4 y el 3
try{
    //lado ab = punto4, p
    a = punto4; b = p; c = punto3;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }

    //lado ab = punto3, p
    a = punto3; b = p; c = punto4;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }
}
catch (Exception &e){
    test = 0;};

//veo el angulo entre el punto 3 y el 1
try{
    //lado ab = punto3, p
    a = punto3; b = p; c = punto1;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }

    //lado ab = punto1, p
    a = punto1; b = p; c = punto3;
    divisor = 2.0 * b->distanciaA(c) * a->distanciaA(c);
    if (divisor != 0)
        test = (pow(b->distanciaA(c), 2) + pow(a->distanciaA(c), 2) - pow(a->distanciaA(b),
2) ) / divisor;
    else
        //el punto p esta sobre la recta. Con este no calculo nada y veo el resto de los
lados
        test = 1;
    if((test > 1) || (test < 0)){
        ok = false; //el angulo es mas mayor a 90 grados
    }
}
catch (Exception &e){
    test = 0;};

return ok;
}

Punto *Cuadrado::centro(){

```



```

    return miCentro;
}

Punto *Cuadrado::getPunto1(){
    return punto1;
}

Punto *Cuadrado::getPunto2(){
    return punto2;
}

Punto *Cuadrado::getPunto3(){
    return punto3;
}

Punto *Cuadrado::getPunto4(){
    return punto4;
}

long double Cuadrado::productoVectorial(Punto *p1, Punto *p2){
    return (p1->x * p2->y) - (p1->y * p2->x);
}

long double Cuadrado::modulo(Punto *p1){
    return sqrt(p1->x * p1->x + p1->y * p1->y);
}

void Perinola::init(int x1, int y1, int x2, int y2){
    rectangulo.Left = x1;
    rectangulo.Right = x2;
    rectangulo.Top = y1;
    rectangulo.Bottom = y2;

    perinolas = new OrderedCollection();
}

void Perinola::getPunto(double *x, double *y){
    if(perinolas->count > 0){
        ((Perinola*)(perinolas->items[random(perinolas->count)]))->getPunto(x, y);
    }
    else{
        *x = random(rectangulo.Right - rectangulo.Left) + rectangulo.Left;
        *y = random(rectangulo.Top - rectangulo.Bottom) + rectangulo.Bottom;
    }
}

void Perinola::eliminarArea(int x1, int y1, int x2, int y2){
    int limite = x2-x1;
    Perinola *p;

    if(perinolas->count > 0)
        for(int x = 0; x < perinolas->count; x++){
            ((Perinola*)(perinolas->items[x]))->eliminarArea(x1, x2, y1, y2);
        }
    else{
        if(((x1 - rectangulo.Left) > limite) && ((y1 - rectangulo.Top) > limite)){
            p = new Perinola();
            p->init(rectangulo.Left, rectangulo.Top, x1, y1);
            perinolas->add(p);
        }
        if((x1 > rectangulo.Left) && (x2 < rectangulo.Right) && ((y1 - rectangulo.top) >
limite)){
            p = new Perinola();
            p->init(x1, rectangulo.Top, x2, y1);
            perinolas->add(p);
        }
        if(((rectangulo.Right - x2) > limite) && ((y1 - rectangulo.Top) > limite)){
            p = new Perinola();
            p->init(x2, rectangulo.Top, rectangulo.Right, y1);
            perinolas->add(p);
        }
    }
}

```

```

        if((y1 > rectangulo.Top) && (y2 < rectangulo.Bottom) && ((x1 - rectangulo.left) >
limite)){
            p = new Perinola();
            p->init(rectangulo.left, y1, x1, y2);
            perinolas->add(p);
        }
        if((y1 > rectangulo.Top) && (y2 < rectangulo.Bottom) && ((rectangulo.Right - x2) >
limite)){
            p = new Perinola();
            p->init(x2, y1, rectangulo.Right, y2);
            perinolas->add(p);
        }
        if(((x1 - rectangulo.Left) > limite) && ((rectangulo.Bottom - y2) > limite)){
            p = new Perinola();
            p->init(rectangulo.Left, y2, x1, rectangulo.Bottom);
            perinolas->add(p);
        }
        if((x1 > rectangulo.Left) && (x2 < rectangulo.Right) && ((rectangulo.Bottom - y2) >
limite)){
            p = new Perinola();
            p->init(x1, y2, x2, rectangulo.Bottom);
            perinolas->add(p);
        }
        if(((rectangulo.Right - x2) > limite) && ((rectangulo.Bottom - y2) > limite)){
            p = new Perinola();
            p->init(x2, y2, rectangulo.Right, rectangulo.Bottom);
            perinolas->add(p);
        }
    }
}

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    archivoTrabajo == NULL;
}
//-----

double TForm1::delimitarAreaConCirculos(int n, double *coordX, double *coordY,
Graphics::TBitmap *grafica){

    double radio = 0, distancia, x1, x2, y1, y2;
    int indicel, indice2;
    bool ok = true;

    for(int x = 0; x < n; x++)
        if(coordX[x] > radio)
            radio = coordX[x];

    while(ok){
        for(int x = 0; (x < (n-1)) && ok; x++){
            for(int y = x + 1; (y < n) && ok; y++){
                distancia = sqrt(pow1((coordX[x] - coordX[y]), 2) + pow1(coordY[x] - coordY[y],
2));
                if (distancia > (radio * 2)){
                    ok = false;
                    indicel = x;
                    indice2 = y;
                }
            }
        }

        //reviso todos los puntos a ver si hay algun punto que este entre el cuadrante
        formado por los puntos x e y
        for(int w = 0; (w < n) && ! ok; w++){
            if((w != x) && (w!= y)){
                //formo un rectangulo con los puntos x e y como extremos.
                //si hay arboles dentro del rectangulo entonces no hay separacion
                x1 = (coordX[x] < coordX[y])?coordX[x]:coordX[y];
                x2 = (coordX[x] < coordX[y])?coordX[y]:coordX[x];
                y1 = (coordY[x] < coordY[y])?coordY[x]:coordY[y];
                y2 = (coordY[x] < coordY[y])?coordY[y]:coordY[x];
            }
        }
    }
}

```

```

        if((coordX[w] > x1) && (coordX[w] < x2) && (coordY[w] > y1) && (coordY[w] <
y2))
            ok = true;

            //si hay un circulo que intersecta los dos circulos que se separan entonces
            //no hay separacion
            x1 = sqrt(powl((coordX[w] - coordX[x]), 2) + powl(coordY[w] - coordY[x],
2));
            x2 = sqrt(powl((coordX[w] - coordX[y]), 2) + powl(coordY[w] - coordY[y],
2));
            if((x1 < (radio *2)) && (x2 < (radio *2)) )
                ok = true;

            //armo un rectangulo entre los circulos x e y
            double factor = 0.7;
            x1 = (coordX[x] < coordX[y])?(coordX[x]-radio*factor):(coordX[y]-
radio*factor);
            x2 = (coordX[x] <
coordX[y])?(coordX[y]+radio*factor):(coordX[x]+radio*factor);
            y1 = (coordY[x] < coordY[y])?(coordY[x]-radio*factor):(coordY[y]-
radio*factor);
            y2 = (coordY[x] <
coordY[y])?(coordY[y]+radio*factor):(coordY[x]+radio*factor);
            if((coordX[w] > x1) && (coordX[w] < x2) && (coordY[w] > y1) && (coordY[w] <
y2))
                ok = true;
        }
    }
    radio-= 1;
}

/*if((CheckBoxDelimitarArea->Checked) && ! ok){
    grafica->Canvas->Pen->Color = clBlue;
    grafica->Canvas->Pen->Width = 2;
    grafica->Canvas->PenPos = Point(coordX[indice1] / 10, grafica->Height -
coordY[indice1] / 10);
    grafica->Canvas->LineTo(coordX[indice2] / 10, grafica->Height - coordY[indice2] / 10);

    grafica->Canvas->Brush->Style = bsClear;
    grafica->Canvas->Ellipse((coordX[indice1] - radio) / 10, grafica->Height -
((coordY[indice1] - radio) / 10), (coordX[indice1] + radio) / 10, grafica->Height -
((coordY[indice1] + radio) / 10));
    grafica->Canvas->Ellipse((coordX[indice2] - radio) / 10, grafica->Height -
((coordY[indice2] - radio) / 10), (coordX[indice2] + radio) / 10, grafica->Height -
((coordY[indice2] + radio) / 10));
}*/

    return radio;
}

void TForm1::dibujarCirculos(int n, double *coordX, double *coordY, Graphics::TBitmap
*grafica, double radio){

    grafica->Canvas->Pen->Color = 0xEEEEEE;
    grafica->Canvas->Pen->Width = 2;
    grafica->Canvas->Brush->Style = bsClear;
    grafica->Canvas->Brush->Color = 0xEEEEEE;
    for(int y = 0; y < n; y++){
        grafica->Canvas->Ellipse((coordX[y] - radio) / 10, grafica->Height - ((coordY[y] -
radio) / 10), (coordX[y] + radio) / 10, grafica->Height - ((coordY[y] + radio) / 10));
    }
}

void TForm1::dibujarMuestras(int n, double *coordX, double *coordY, Graphics::TBitmap
*grafica){
    grafica->Canvas->Pen->Color = clRed;
    grafica->Canvas->Pen->Width = 3;

    for(int y = 0; y < n; y++){

```

```

        grafica->Canvas->PenPos = Point(coordX[y] / 10 -1, grafica->Height - coordY[y] / 10 -
1);
        grafica->Canvas->LineTo(coordX[y] / 10 + 1, grafica->Height - coordY[y] / 10 + 1);
    }
}

void TForm1::cuadratasAlAzar(){
    AnsiString linea, figura = "cuadrado";
    long double lado1, angulo, angulo2;
    int cuadratas, cantidadCaquitas, mosaico2, cantidadMosaico, cantidadMosaico2, indicex,
        cantidad;
    Punto *punto1 = new Punto(), *punto2 = new Punto(),
        *punto3 = new Punto(), *punto4 = new Punto(), *punto5 = new Punto();
    Graphics::TBitmap *laguna = new Graphics::TBitmap();
    bool ok, flag, cancelar = false;
    TablaDeFrecuencia *frecuencia = new TablaDeFrecuencia();
    OrderedCollection *cuadrados = new OrderedCollection();
    Figura *cuad = new Cuadrado(), *cuad2;
    long double xmin, xmax, ymin, ymax, coordX[100], coordY[100];
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r"),
        *salida;
    char *buffer = (char*)malloc(sizeof(char) * 100);
    PuntoMapeado *mosaico, *indiceMosaico, *indiceDeX;
    double offsetX = 16000;
    double offsetY = 15000;

    cantidad = 0;
    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    xmax = xmin = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1)) + offsetX;
    ymax = ymin = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000)) + offsetY;
    while(! feof(entrada)){
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            coordX[cantidad] += offsetX;
            if(coordX[cantidad] > xmax)
                xmax = coordX[cantidad];
            if(coordX[cantidad] < xmin)
                xmin = coordX[cantidad];
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
            coordY[cantidad] += offsetY;
            if(coordY[cantidad] > ymax)
                ymax = coordY[cantidad];
            if(coordY[cantidad] < ymin)
                ymin = coordY[cantidad];
            cantidad++;
        }

        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
    }
    fclose(entrada);

    xmax += offsetX;
    ymax += offsetY;

    //armo el mosaico
    int paso = 1;
    while(((xmax - xmin) / paso) > 500)
        paso++;
    while(((ymax - ymin) / paso) > 500)
        paso++;
    int longitudMosaico = ((xmax - xmin + 1) / paso) * ((ymax - ymin + 1) / paso);
    while(longitudMosaico < 0){
        paso++;
        longitudMosaico = ((xmax - xmin + 1) / paso) * ((ymax - ymin + 1) / paso);
    }
    mosaico = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (longitudMosaico + 1));
    indiceMosaico = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (longitudMosaico + 1));
    indiceDeX = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (((xmax - xmin + 1) / 2) + 1));

```

```

        //indiceDeX guarda las posiciones del vector mosaico donde comienzan las
columnas de la matriz
        laguna->Height = ymax / paso * 2;
        laguna->Width = xmax / paso * 2 + 55;

        //dibujo los individuos
        for(int x = 0; x < cantidad; x++){
            laguna->Canvas->Pen->Color = clRed;
            laguna->Canvas->Pen->Width = 3;
            laguna->Canvas->PenPos = Point(coordX[x] / paso * 2, laguna->Height - coordY[x] / paso
* 2);
            laguna->Canvas->LineTo(coordX[x] / paso * 2 + 1, laguna->Height - coordY[x] / paso * 2
+ 1);
            //laguna->Canvas->TextOut(coordX[x] / paso * 2, laguna->Height - coordY[x] / paso * 2,
IntToStr(x));
        }

        //dibujo el perímetro
        laguna->Canvas->Pen->Color = clBlue;
        laguna->Canvas->Pen->Width = 2;
        laguna->Canvas->PenPos = Point((124958.9144 + offsetX) / paso * 2, laguna->Height - (-
14197.68422 + offsetY) / paso * 2);
        laguna->Canvas->LineTo((72087.06114 + offsetX) / paso * 2, laguna->Height - (122631.077
+ offsetY) / paso * 2);
        laguna->Canvas->LineTo((34251.46867 + offsetX) / paso * 2, laguna->Height - (96137.4043
+ offsetY) / paso * 2);
        laguna->Canvas->LineTo((-15701.82424 + offsetX) / paso * 2, laguna->Height -
(11682.39416 + offsetY) / paso * 2);
        laguna->Canvas->LineTo((124958.9144 + offsetX) / paso * 2, laguna->Height - (-
14197.68422 + offsetY) / paso * 2);

        //inicializo el mapeo de puntos
        cantidadMosaico2 = 0;
        indicex = 0;
        for(int x = xmin; x <= xmax; x+= paso){
            indiceDeX[indicex].x = x;
            indiceDeX[indicex].indice = cantidadMosaico2;
            indicex++;
            for(int y = ymin; y <= ymax; y+= paso){
                puntol->x = x;
                puntol->y = y;
                if(this->estaEnElAreaDeMuestreo(puntol, xmax, ymax)){
                    mosaico[cantidadMosaico2].x = x;
                    mosaico[cantidadMosaico2].y = y;
                    mosaico[cantidadMosaico2].indice = cantidadMosaico2;
                    indiceMosaico[cantidadMosaico2].indice = cantidadMosaico2;
                    cantidadMosaico2++;
                }
            }
        }
        cantidadMosaico = cantidadMosaico2;
        indiceDeX[indicex].x = -1;

        //inicializo variables
        frecuencia->vaciar();
        cuadrados->clear();
        flag = true;

        //inicializo el lado del cuadrado y la cantidad de cuadratas
        lado1 = StrToFloat(EditTamanoDeLado->Text);
        cuadratas = StrToFloat(EditNCuadratas->Text);
        int corte = 0;
        int nreal = cuadratas;

        randomize();

        cantidadMosaico = cantidadMosaico2;
        for(int y = 0; y < cantidadMosaico; y++)
            indiceMosaico[y].x = 0;

        ProgressBar1->Position = 0;

```

```

ProgressBar1->Max = cuadratas;
ProgressBar2->Position = 0;
ProgressBar2->Max = cantidadMosaico;
for(int y = 1; (y <= cuadratas) && flag; y++){
    ProgressBar1->StepIt();
    Label8->Caption = "Colocando cuadrata no. " + IntToStr(y) + "    ";
    Label8->Repaint();

    ok = true;
    while(ok){
        if (cancelar)
            return;
        ProgressBar2->StepIt();
        Label8->Repaint();
        Label9->Repaint();

        //armar el cuadrado dentro del area de muestreo
        //elijo el punto1
        //Para esto elijo un mosaico al azar

        mosaico2 = random(cantidadMosaico);
        flag = true;

        punto1->x = mosaico[indiceMosaico[mosaico2].indice].x;
        punto1->y = mosaico[indiceMosaico[mosaico2].indice].y;

        while(flag && ! this->estaEnElAreaDeMuestreo(punto1, xmax, ymax)){
            //elimino el mosaico 'mosaico2'
            indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
            indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
            indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
            //actualizo el indice en mosaico
            mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
            mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
            cantidadMosaico--;

            mosaico2 = random(cantidadMosaico);
            if(cantidadMosaico == 0)
                flag = false;
            flag = cantidadMosaico != 0;
            punto1->x = mosaico[indiceMosaico[mosaico2].indice].x;
            punto1->y = mosaico[indiceMosaico[mosaico2].indice].y;
        }

        //elijo el punto2 (a partir de un angulo)
        angulo = random(90); //random(360);

        for(int rotaciones = 1; rotaciones < 90; rotaciones+= 5){

            if(angulo < 90){
                punto2->x = punto1->x + cosl(angulo * pi / 180) * lado1;
                punto2->y = punto1->y + sinl(angulo * pi / 180) * lado1;
            }
            else
                if(angulo < 180){
                    angulo2 = 180 - angulo;
                    punto2->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
                    punto2->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
                }
                else
                    if(angulo < 270){
                        angulo2 = angulo - 180;
                        punto2->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
                        punto2->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                    }
                    else{
                        angulo2 = 360 - angulo;
                        punto2->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                        punto2->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                    }
        }
    }
}

```

```

//si el punto2 no esta en el area de muestreo lo giro 180 grados tomando
como eje el punto x1
if(! this->estaEnElAreaDeMuestreo(punto2, xmax, ymax)){
    punto2->x = punto1->x - (punto2->x - punto1->x);
    punto2->y = punto1->y - (punto2->y - punto1->y);
}
//calculo los punto3 y punto4

if(punto1->x == punto2->x){
    punto3->x = punto4->x = punto1->x - lado1;
    punto3->y = punto1->y;
    punto4->y = punto2->y;
    if((! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax)) || (! this-
>estaEnElAreaDeMuestreo(punto4, xmax, ymax))){
        //volteo el cuadrado tomando el eje punto1-punto2
        punto3->x = punto4->x = punto1->x + lado1;
    }
}
else
    if(punto1->x < punto2->x){
        if(angulo < 180){
            angulo2 = 90 - angulo;
            punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
            if((! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax)) || (! this-
>estaEnElAreaDeMuestreo(punto4, xmax, ymax))){
                //volteo el cuadrado tomando el eje punto1-punto2
                punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
            }
        }
        else{
            angulo2 = angulo - 270;
            punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
            if((! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax)) || (!
this->estaEnElAreaDeMuestreo(punto4, xmax, ymax))){
                //volteo el cuadrado tomando el eje punto1-punto2
                punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
            }
        }
    }
}
else{
    if(angulo < 90){
        angulo2 = 90 - angulo;
        punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
        punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
        punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
        punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
        if((! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax)) || (!
this->estaEnElAreaDeMuestreo(punto4, xmax, ymax))){
            //volteo el cuadrado tomando el eje punto1-punto2
            punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
        }
    }
    else{
        angulo2 = 270 - angulo;
        punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
        punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
    }
}

```

```

        punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
        punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
        if(! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax)) || (!
this->estaEnElAreaDeMuestreo(punto4, xmax, ymax)){
            //volteo el cuadrado tomando el eje punto1-punto2
            punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
        }
    }

    ok = false;
    ((Cuadrado*)cuad)->setPuntos(punto1, punto2, punto3, punto4);
    for(int ww = 0; ww < cuadrados->count; ww++){
        if( ((Cuadrado*)cuad)->intersectasCon((Cuadrado*)(cuadrados->items[ww])) )
            ok = true;
    }
    // if(ok){
    //     Cuadrado *nuevo = new Cuadrado(punto1, punto2, punto3, punto4);
    //     cuadrados->add(nuevo);
    // }

    /*int maxx = max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)), maxy
= max(max(punto1->y, punto2->y), max(punto3->y, punto4->y));
    int minx = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)), miny =
min(min(punto1->y, punto2->y), min(punto3->y, punto4->y));
    minx-= xmin; miny-= ymin;
    minx = minx / paso; miny = miny / paso;
    minx = minx * paso + xmin; miny = miny * paso + ymin;
    for(int ww = minx; (ww <= maxx) && ! ok; ww+= paso){
        //busco en el indice de x
        indicex = 0;
        while((indiceDeX[indicex].x != ww) && (indiceDeX[indicex].x != -1))
            indicex++;
        indicex = indiceDeX[indicex].indice;
        for(int zz = miny; (zz <= maxy) && ! ok; zz+= paso){
            punto5->x = ww;
            punto5->y = zz;
            if(cuad->tePerteneceElPunto(punto5)){
                //busco en mosaico el indice de y
                while((mosaico[indicex].y != zz) && (mosaico[indicex].x == ww))
indicex++;

                if(mosaico[indicex].x == ww){
                    ok = (indiceMosaico[mosaico[indicex].indice].x == 1);
                }
            }
        }
    }*/
    angulo = int(angulo + 5) % 90;

    if(! ok)
        rotaciones = 90;
    } //del for rotaciones

    //ok = subarea->intersectaAlgunAreaAnalizada((Cuadrado*)cuad = new
Cuadrado(punto1, punto2, punto3, punto4));
    ok = ok || ! this->estaEnElAreaDeMuestreo(punto1, xmax, ymax);
    ok = ok || ! this->estaEnElAreaDeMuestreo(punto2, xmax, ymax);
    ok = ok || ! this->estaEnElAreaDeMuestreo(punto3, xmax, ymax);
    ok = ok || ! this->estaEnElAreaDeMuestreo(punto4, xmax, ymax);

    if(ok){
        //si luego de 20 intentos de armar un cuadrado a partir de
indiceMosaico[mosaico2] no lo logré
        //elimino el mosaico.
        indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
        indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
        indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
        //actualizo el indice en mosaico

```



```

mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
cantidadMosaico--;

if(cantidadMosaico == 0)
    flag = false;
}

if(cantidadMosaico == 0)
    flag = false;
flag = cantidadMosaico != 0;
if(! flag)
    ok=false; //salgo a la fuerza. No hay lugar para poner cuadrados

corte++;
Label9->Caption = "Corte " + IntToStr(corte) + " ";
Label9->Repaint();
if(corte == 500){
    ok = MessageDlg("Seguir con otros 10000?", mtConfirmation, botonesSiNo, 0)
== mrYes;

    if(ok)
        corte = 0;
    else{
        nreal = y - 1;
        y = cuadratas;
    }
}
} //del while ok

if(flag){
    //elimino el mosaico 'mosaico2'
    //mosaico[mosaico2].x = -1;
    indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
    indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
    indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
    //actualizo el indice en mosaico
    mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
    mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
    cantidadMosaico--;
    if(cantidadMosaico == 0)
        flag = false;

    cuadrados->add(cuad2 = new Cuadrado(punto1, punto2, punto3, punto4));

    //eliminar los mosaicos que estan dentro del cuadrado
    ok = true;
    int maxx = max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)), maxy =
max(max(punto1->y, punto2->y), max(punto3->y, punto4->y));
    int minx = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)), miny =
min(min(punto1->y, punto2->y), min(punto3->y, punto4->y));
    minx-= xmin; miny-= ymin;
    minx = minx / paso; miny = miny / paso;
    minx = minx * paso + xmin; miny = miny * paso + ymin;
    for(int ww = minx; ww <= maxx; ww+= paso)
        for(int zz = miny; zz <= maxy; zz+= paso){
            punto5->x = ww;
            punto5->y = zz;
            if(cuad2->tePerteneceElPunto(punto5)){
                //busco en el indice de x
                indicex = 0;
                while((indiceDeX[indicex].x != ww) && (indiceDeX[indicex].x != -1))
                    indicex++;
                indicex = indiceDeX[indicex].indice;
                //busco en mosaico el indice de y
                while((mosaico[indicex].y != zz) && (mosaico[indicex].x == ww) &&
(indicex < longitudMosaico))
                    indicex++;
                if(mosaico[indicex].x == ww){
                    mosaico2 = mosaico[indicex].indice;
                    indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
                    indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];

```

```

        indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
        //marco el mosaico como "usado" por un cuadrado
        indiceMosaico[cantidadMosaico].x = 1;
        //actualizo el indice en mosaico
        mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
        mosaico[indiceMosaico[cantidadMosaico].indice].indice =
cantidadMosaico;
        cantidadMosaico--;
        if(cantidadMosaico == 0)
            flag = false;
        ProgressBar2->StepIt();
    }
}

//ver caquitas dentro de area
cantidadCaquitas = 0;
for(int x = 0; x < cantidad; x++){
    punto5->x = coordX[x];
    punto5->y = coordY[x];
    if(cuad2->tePerteneceElPunto(punto5))
        cantidadCaquitas++;
}
if(corte < 500)
    frecuencia->agregar(cantidadCaquitas);

//*****
//Dibujo el perimetro en un bitmap
if((figura == "cuadrado") && (corte < 500)){
    laguna->Canvas->Pen->Color = clBlack;
    laguna->Canvas->Pen->Width = 2;
    laguna->Canvas->PenPos = Point((punto1->x) / paso * 2, laguna->Height -
(punto1->y) / paso * 2);
    laguna->Canvas->LineTo((punto2->x) / paso * 2, laguna->Height - (punto2->y) /
paso * 2);
    laguna->Canvas->LineTo((punto4->x) / paso * 2, laguna->Height - (punto4->y) /
paso * 2);
    laguna->Canvas->LineTo((punto3->x) / paso * 2, laguna->Height - (punto3->y) /
paso * 2);
    laguna->Canvas->LineTo((punto1->x) / paso * 2, laguna->Height - (punto1->y) /
paso * 2);
}

} //del if flag

} //del for de 1 hasta cuadratas
//} //del for de 1 hasta mosaicos

laguna->SaveToFile("Mapa cuadratas al azar.bmp");
laguna->FreeImage();
delete laguna;

if(SaveDialog1->Execute()){
    //escribo la salida
    salida = fopen(SaveDialog1->FileName.c_str(), "w+");
    fprintf(salida, ("N = " + IntToStr(nreal) + "\n").c_str());
    fprintf(salida, ("Tamano de cuadrado: " + FloatToStr(lado1) + " x " +
FloatToStr(lado1) + "\n").c_str());
    fprintf(salida, "\nTabla de frecuencia\n");
    for(int x = 0; x <= frecuencia->cantidadMaxima(); x++)
        fprintf(salida, (IntToStr(x) + "\t" + IntToStr(frecuencia->frecuenciaDe(x)) +
"\n").c_str());
    fclose(salida);
}

free(mosaico);
free(indiceMosaico);
free(indiceDeX);

delete frecuencia;
}

```

```

void TForm1::cuadratasPorGrilla(){
    //leer los datos del archivo
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r"),
        *salida = fopen("salida Byth y Ripley.txt", "w+");
    char *buffer = (char*)malloc(sizeof(char) * 100);
    AnsiString linea;
    TablaDeFrecuencia *tabla = new TablaDeFrecuencia();
    double maxX = 0, maxY = 0, coordX[100], coordY[100], centroX, centroY;
    int cantidad = 0, indice1, indice2, n, lado, cantidadX, cantidadY, contador;
    bool ok, *grilla;

    while(! feof(entrada)){
        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            if(coordX[cantidad] > maxX)
                maxX = coordX[cantidad];
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
            if(coordY[cantidad] > maxY)
                maxY = coordY[cantidad];
            cantidad++;
        }
    }

    maxX+= 100; //limites del area de trabajo
    maxY+= 100; //limites del area de trabajo

    fclose(entrada);
    free(buffer);

    lado = RU_StrToFloat(EditTamanoDeLado->Text);
    cantidadX = (maxX / lado) + 1;
    cantidadY = (maxY / lado) + 1;

    grilla = (bool*)malloc(sizeof(bool) * cantidadX * cantidadY);
    for(int x = 0; x < (cantidadX * cantidadY); x++){
        grilla[x] = false; //seteo todas las cuadratas como no-elegidas
    }

    n = cantidadX * cantidadY;
    /*n = StrToInt(EditNCuadratas->Text);

    if(n > (cantidadX * cantidadY)){
        ShowMessage("El n es mas grande que las cuadratas que forman la grilla (" +
            FloatToStr(cantidadX * cantidadY) + ")");
        return;
    }*/

    randomize();
    for(int x = 0; x < cantidadX; x++){
        for(int yy = 0; yy < cantidadY; yy++){
            ok = false;
            while (! ok){
                //indice1 = random(cantidadX);
                //indice2 = random(cantidadY);
                indice1 = x;
                indice2 = yy;
                if(! grilla[cantidadY * indice1 + indice2]){
                    grilla[cantidadY * indice1 + indice2] = ok = true;
                }
            }
        }
    }

    contador = 0;

    //calculo el centro del cuadrado elegido
    centroX = indice1 * lado + lado / 2;
    centroY = indice2 * lado + lado / 2;

    //hago el conteo de árboles dentro del cuadrado
    for(int y = 0; y < cantidad; y++){

```

```

        if((coordX[y] > (centroX - lado / 2)) &&
            (coordX[y] < (centroX + lado / 2)) &&
            (coordY[y] > (centroY - lado / 2)) &&
            (coordY[y] < (centroY + lado / 2)) )
            contador++;
    }

    tabla->agregar(contador);
}

if(SaveDialog1->Execute()){
    //escribo la salida
    salida = fopen(SaveDialog1->FileName.c_str(), "w+");
    fprintf(salida, ("N = " + IntToStr(n) + "\n").c_str());
    fprintf(salida, ("Tamano de grilla: " + IntToStr(cantidadX) + " x " +
IntToStr(cantidadY) + "\n").c_str());
    fprintf(salida, ("Tamano de cuadrado: " + FloatToStr(lado) + " x " + FloatToStr(lado)
+ "\n").c_str());
    fprintf(salida, "\nTabla de frecuencia\n");
    for(int x = 0; x <= tabla->cantidadMaxima(); x++)
        fprintf(salida, (IntToStr(x) + "\t" + IntToStr(tabla->frecuenciaDe(x)) +
"\n").c_str());
    fclose(salida);
}

delete tabla;
}

bool TForm1::estaEnElAreaDeMuestreo(Punto *p, double x, double y){
    //return (p->x > 0) && (p->y > 0) && (p->x < x) && (p->y < y);
    bool ok = true;
    double offsetX = 16000;
    double offsetY = 15000;

    ok = ok && ((p->y - offsetY) >= (-0.183989354 * (p->x - offsetX) + 8793.425669));
    ok = ok && ((p->y - offsetY) <= (-2.587932004 * (p->x - offsetX) + 309187.4896));
    ok = ok && ((p->y - offsetY) <= (1.690679537 * (p->x - offsetX) + 38229.1471));
    ok = ok && ((p->y - offsetY) <= (0.700231474 * (p->x - offsetX) + 72153.44792));

    return ok;
}

bool TForm1::estaLaRectaDentroDelArea(double x1, double y1, double x2, double y2, double
*coordX, double *coordY, int cantidad, double radio){
    int paso = 10;
    double x, y, distancia, min, max;
    bool ok;

    if (fabs(x1 - x2) > fabs(y1 - y2)){
        if(x1 > x2){
            min = x2;
            max = x1;
        }
        else{
            min = x1;
            max = x2;
        }
        for(int x = min; x <= max; x+= paso){
            y = (x - x1) * (y2 - y1) / (x2 - x1) + y1;
            ok = false;
            for(int z = 0; z < cantidad; z++){
                distancia = sqrt(powl(x - coordX[z], 2) + powl(y - coordY[z], 2));
                if (distancia < radio)
                    ok = true;
            }
            if(! ok)
                return false; //encontre un punto de la recta que no cae dentro de ningun círculo
        }
    }
}

```

```

    }
    else{
        if(y1 > y2){
            min = y2;
            max = y1;
        }
        else{
            min = y1;
            max = y2;
        }
        for(int y = min; y <= max; y+= paso){
            x = (y - y1) * (x2 - x1) / (y2 - y1) + x1;
            ok = false;
            for(int z = 0; z < cantidad; z++){
                distancia = sqrt(pow(x - coordX[z], 2) + pow(y - coordY[z], 2));
                if (distancia < radio)
                    ok = true;
            }
            if(! ok)
                return false; //encontre un punto de la recta que no cae dentro de ningun
círculo
        }
    }

    return true;
}

void __fastcall TForm1::ButtonBythYRipleyClick(TObject *Sender)
{
    //leer los datos del archivo
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r"),
        *salida = fopen("salida Byth y Ripley.txt", "w+"), *circulos;
    char buffer[200];
    AnsiString linea;

    double coordX[100], coordY[100], circulosX[1000], circulosY[1000], swap, distancia,
        maxX = 0, maxY = 0, xx[100], rr[100], cuadradoX[50], radio,
        cuadradoY[50];
    int cantidad = 0, indice1, indice2, n, tamanoCuadrado, cantidadCirculos;
    bool ok;

    while(! feof(entrada)){
        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            if(coordX[cantidad] > maxX)
                maxX = coordX[cantidad];
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
            if(coordY[cantidad] > maxY)
                maxY = coordY[cantidad];
            cantidad++;
        }
    }

    maxX+= 100; //limites del area de trabajo
    maxY+= 100; //limites del area de trabajo

    fclose(entrada);
    //free(buffer);

    //ordenar aleatoriamente los valores
    randomize();
    /*for(int x = 1; x <= 500; x++){
        indice1 = random(cantidad - 1);
        indice2 = random(cantidad - 1);

        swap = coordX[indice1];
        coordX[indice1] = coordX[indice2];
        coordX[indice2] = swap;
    }

```

```

        swap = coordY[indice1];
        coordY[indice1] = coordY[indice2];
        coordY[indice2] = swap;
    }*/

n = StrToInt(EditN->Text);

//Metodo de Byth & Ripley. Punto 4.2.1 de Ecological Methodology. CJ Krebs
//busco n puntos al azar y mido la distancia al punto mas cercano
for(int x = 0; x < n; x++){
    indice1 = random(maxX - 1);
    indice2 = random(maxY - 1);
    xx[x] = maxX;

    for (int y = 0; y < cantidad; y++){
        distancia = sqrt(pow((indice1 - coordX[y]), 2) + pow(indice2 - coordY[y], 2));
        if(distancia < xx[x])
            xx[x] = distancia;
    }
}

double LIMITE = StrToInt(EditLimiteCuadrado->Text);

cantidadCirculos = 0;
if(CheckBoxArchivoDeCirculos->Checked){
    //los centros de los circulos son leidos de un archivo
    //guardo en coordX y coordY esos centros, guardo en cantidaCirculos la cantidad
    //de circulos leidos
    circulos = fopen(EditArchivoDeCirculos->Text.c_str(), "r");
    while(! feof(circulos)){
        RU_LeerLinea(circulos, buffer);
        linea = AnsiString(buffer);
        if(linea.Length() > 0){
            circulosX[cantidadCirculos] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";")
- 1));
            circulosY[cantidadCirculos] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") +
1, 1000));
            cantidadCirculos++;
        }
    }
    fclose(circulos);
}

if(CheckBoxDelimitarArea->Checked){
    if(RadioButtonOptimo->Checked)
        ShowMessage("El procedimiento de radio óptimo para Byth & Ripley todavía no está
progradado.");
    else{
        radio = RU_StrToFloat(EditRadio->Text);
    }
}

int listo = 1;
while(listo > 0){

    Label9->Caption = IntToStr(listo) + "          ";
    Label9->Repaint();

    //busco n puntos al azar, creo cuadrados y cuento los arboles
    //supongo que un cudrado no crecerá mas de LIMITE unidades, por eso
    //"corto" los limites en esa cantidad
    //perinola->init(0 + LIMITE, 0 + LIMITE, maxX - LIMITE, maxY - LIMITE);
    indice2 = 0;
    LIMITE = StrToInt(EditLimiteCuadrado->Text);
    for(int x = 0; x < n; x++){
        ok = true;
        indice1 = 0;
        while(ok){
            //cuadradoX[x] = random(maxX - LIMITE*2) + LIMITE;
            //cuadradoY[x] = random(maxY - LIMITE*2) + LIMITE;
            cuadradoX[x] = random(maxX ) ;

```

```

cuadradoY[x] = random(maxY ) ;

ok= false;
//veo si el punto elegido al azar cayó dentro de un circulo
if(CheckBoxDelimitarArea->Checked){
    ok = true;
    for (int y = 0; (y < cantidadCirculos) && ok; y++){
        distancia = sqrt(powl((cuadradoX[x] - circulosX[y]), 2) + powl(cuadradoY[x] -
circulosY[y], 2));
        if(distancia < radio)
            ok = false; //si detecto que el punto cayo dentro de un círculo, pongo ok en
false para chequear
                                //si se superpone con otro cuadrado.
    }

    /*Controlo que los cuatro vertices esten dentro de los círculos que limitan el
area
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[x] - LIMITE - circulosX[y]), 2) +
powl(cuadradoY[x] - LIMITE - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[x] - LIMITE - circulosX[y]), 2) +
powl(cuadradoY[x] + LIMITE - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[x] + LIMITE - circulosX[y]), 2) +
powl(cuadradoY[x] - LIMITE - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[x] + LIMITE - circulosX[y]), 2) +
powl(cuadradoY[x] + LIMITE - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
}*/
}

//controlo que no se solape con otro cuadrado ya creado, ni con su
//area de crecimiento
for(int y = 0; y < x; y++){
    if( (fabs(cuadradoX[x] - cuadradoX[y]) < LIMITE * 2) &&
        (fabs(cuadradoY[x] - cuadradoY[y]) < LIMITE * 2) )
        ok = true;
}

indice1++;
if(indice1 == 80000){
    indice2++;
    if (indice2 == 100){
        ShowMessage("Imposible crear los cuadrados");
        Graphics::TBitmap *grafica = new Graphics::TBitmap();
        grafica->Width = maxX / 10;
        grafica->Height = maxY / 10;
    }
}

```

```

        if(CheckBoxDelimitarArea->Checked)
            this->dibujarCirculos(cantidadCirculos, circulosX, circulosY, grafica, radio);
        this->dibujarMuestras(cantidad, coordX, coordY, grafica);

        grafica->Canvas->Pen->Color = clBlack;
        grafica->Canvas->Pen->Width = 2;
        for(int y = 0; y < x; y++){
            grafica->Canvas->PenPos = Point((cuadradoX[y] - LIMITE) / 10, grafica->Height -
            (cuadradoY[y] - LIMITE) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + LIMITE) / 10, grafica->Height -
            (cuadradoY[y] - LIMITE) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + LIMITE) / 10, grafica->Height -
            (cuadradoY[y] + LIMITE) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - LIMITE) / 10, grafica->Height -
            (cuadradoY[y] + LIMITE) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - LIMITE) / 10, grafica->Height -
            (cuadradoY[y] - LIMITE) / 10);
        }
        grafica->SaveToFile("mapa muestreo Byth y Ripley.bmp");
        delete grafica;
        return;
    }
    x = 0;
    ok = false;
}

}

//calculo el maximo limite de crecimiento
LIMITE = maxX;
distancia = maxX;
for (int x = 0; x < n; x++)
    for (int z = 0; z < n; z++)
        if (x != z){
            indicel = sqrt(powl((cuadradoX[x] - cuadradoX[z]), 2) + powl(cuadradoY[x] -
            cuadradoY[z], 2));
            if(indicel < distancia){
                distancia = indicel;
                LIMITE = (fabs(cuadradoX[x] - cuadradoX[z]) > fabs(cuadradoY[x] -
            cuadradoY[z]))?(fabs(cuadradoX[x] - cuadradoX[z]):(fabs(cuadradoY[x] - cuadradoY[z]));
            }
        }
    LIMITE = LIMITE / 2;

//hago crecer los cuadrados hasta tener en promedio p arboles (p dado por pantalla)
indicel = 0;
ok = false;
tamanioCuadrado = 0;
listo++;
while(((indicel / n) < StrToInt(EditPromedio->Text)) && (indicel >= 0) && ! ok){
    tamanioCuadrado+= 10;
    if (tamanioCuadrado > LIMITE){
        ok = true;
    }

    indicel = 0;
    /*
    for(int z = 0; (z < n) && (indicel >= 0) && CheckBoxDelimitarArea->Checked && ! ok;
    z++){
        //Controlo que los cuatro vertices del cuadrado esten dentro de los círculos que
        limitan el area
        ok = CheckBoxDelimitarArea->Checked;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[z] - tamanioCuadrado - circulosX[y]), 2) +
            powl(cuadradoY[z] - tamanioCuadrado - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
        if(! ok){
            ok = true;

```



```

        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[z] - tamanoCuadrado - circulosX[y]), 2) +
                powl(cuadradoY[z] + tamanoCuadrado - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[z] + tamanoCuadrado - circulosX[y]), 2) +
                powl(cuadradoY[z] - tamanoCuadrado - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(! ok){
        ok = true;
        for (int y = 0; (y < cantidadCirculos) && ok; y++){
            distancia = sqrt(powl((cuadradoX[z] + tamanoCuadrado - circulosX[y]), 2) +
                powl(cuadradoY[z] + tamanoCuadrado - circulosY[y], 2));
            if(distancia < radio)
                ok = false;
        }
    }
    if(ok)
        indicel-= 100; //el cuadrado z no esta completamente dentro de los circulos.
                        //resto 100 a indice 1 para que no se cumpla la condición de corte,
y así comience de nuevo
    } */

//controlo que no se solape con otro cuadrado ya creado, ni con su area de crecimiento
for(int z = 0; (z < (n-1)) && (indicel >= 0); z++){
    for(int y = z + 1; (y < n) && (indicel >= 0); y++){
        if( (fabs(cuadradoX[z] - cuadradoX[y]) < tamanoCuadrado * 2) &&
            (fabs(cuadradoY[z] - cuadradoY[y]) < tamanoCuadrado * 2) )
            indicel-= 100;
    }
}

//hago el conteo de árboles dentro de los cuadrados
for(int x = 0; (x < cantidad) && (indicel >= 0); x++){
    for(int z = 0; (z < n) && (indicel >= 0); z++){
        if((coordX[x] > (cuadradoX[z] - tamanoCuadrado)) &&
            (coordX[x] < (cuadradoX[z] + tamanoCuadrado)) &&
            (coordY[x] > (cuadradoY[z] - tamanoCuadrado)) &&
            (coordY[x] < (cuadradoY[z] + tamanoCuadrado)) )
            indicel++;
    }
}
if(indicel < 0)
    Label8->Caption = FloatToStr((indicel+100) / n) + " ";
else
    Label8->Caption = FloatToStr(indicel / n) + " ";
Label8->Repaint();
}

if ((indicel / n) >= StrToInt(EditPromedio->Text))
    listo = 0;

if(listo > 500){
    ShowMessage("Error. Los cuadrados crecieron demasiado.");
    Graphics::TBitmap *grafica = new Graphics::TBitmap();
    grafica->Width = maxX / 10;
    grafica->Height = maxY / 10;

    if(CheckBoxDelimitarArea->Checked)
        this->dibujarCirculos(cantidadCirculos, circulosX, circulosY, grafica, radio);
    this->dibujarMuestras(cantidad, coordX, coordY, grafica);

    grafica->Canvas->Pen->Color = clBlack;
}

```

```

        grafica->Canvas->Pen->Width = 2;
        for(int y = 0; y < n; y++){
            grafica->Canvas->PenPos = Point((cuadradoX[y] - tamanoCuadrado) / 10, grafica-
>Height - (cuadradoY[y] - tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + tamanoCuadrado) / 10, grafica->Height -
(cuadradoY[y] - tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + tamanoCuadrado) / 10, grafica->Height -
(cuadradoY[y] + tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - tamanoCuadrado) / 10, grafica->Height -
(cuadradoY[y] + tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - tamanoCuadrado) / 10, grafica->Height -
(cuadradoY[y] - tamanoCuadrado) / 10);
        }
        grafica->SaveToFile("mapa muestreo Byth y Ripley.bmp");
        delete grafica;
        return;
    }

} //del while listo > 0

//calculo el vecino mas cercano de los primeros n arboles que estan dentro
//de cuadrados
indice1 = 0;
for(int x = 0; x < n;){
    ok = false;
    for(int y = 0; y < n; y++){
        if((coordX[indice1] > (cuadradoX[y] - tamanoCuadrado)) &&
(coordX[indice1] < (cuadradoX[y] + tamanoCuadrado)) &&
(coordY[indice1] > (cuadradoY[y] - tamanoCuadrado)) &&
(coordY[indice1] < (cuadradoY[y] + tamanoCuadrado)) ){

            rr[x] = maxX;
            for (int z = 0; z < cantidad; z++){
                distancia = sqrt(pow((coordX[indice1] - coordX[z]), 2) + pow((coordY[indice1] -
coordY[z]), 2));
                if( (distancia < rr[x]) && (indice1 != z))
                    rr[x] = distancia;
            }
            x++;
        }
    }

    indice1++;
}

//-----
/*n = 17;
FILE *temporal = fopen("temp.txt", "r");
for(int x = 0; x < n; x++){
    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    xx[x] = StrToFloat(linea);

    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    rr[x] = StrToFloat(linea);
}
fclose(temporal);*/

if(SaveDialog1->Execute()){
    //escribo la salida
    salida = fopen(SaveDialog1->FileName.c_str(), "w+");
    //escribo en la salida
    fprintf(salida, ("N = " + IntToStr(n) + "\n").c_str());

    double factor = StrToFloat(EditFactor->Text);

    double acum1 = 0;
    double acum2 = 0;
    fprintf(salida, "i\txi\ttri\n");
    for(int x = 0; x < n; x++){

```

```

        fprintf(salida, (IntToStr(x+1) + "\t" + FloatToStr(xx[x]) + "\t" + FloatToStr(rr[x])
+ "\n").c_str());
        acum1+= xx[x] * xx[x];
        acum2+= rr[x] * rr[x];
    }
    fprintf(salida, "\n");
    double h = acum1 / acum2;
    fprintf(salida, "Test de Hopkins (4.9)\n");
    fprintf(salida, ("h = " + FloatToStrF(h, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Indice del Test de Hopkins (4.10)\n");
    fprintf(salida, ("Ih = " + FloatToStrF(h / (1 + h), ffFixed, 3, 3) + "\n").c_str());
    double N1 = n / (pi * acum1) * factor;
    fprintf(salida, "Estimado de la densidad (4.11)\n");
    fprintf(salida, "(Punto al azar al organismo mas cercano)\n");
    fprintf(salida, ("N1 = " + FloatToStrF(N1, ffFixed, 3, 3) + "\n").c_str());
    double varN1 = (1 / N1) * (1 / N1) / n;
    double se = sqrt(varN1 / n);
    fprintf(salida, ("SE(N1) = " + FloatToStrF(se, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Bandas de confianza al 95 %\n");
    fprintf(salida, (FloatToStrF( 1 / (1 / N1 + TDeStudent(n-1) * se), ffFixed, 3, 3) +
"\t" + FloatToStrF( 1 / (1 / N1 - TDeStudent(n-1) * se), ffFixed, 3, 3) + "\n").c_str());
    double N2 = n / (pi * acum2) * factor;
    fprintf(salida, "Estimado de la densidad (4.12)\n");
    fprintf(salida, "(De un organismo a su vecino mas cercano)\n");
    fprintf(salida, ("N2 = " + FloatToStrF(N2, ffFixed, 3, 3) + "\n").c_str());
    double varN2 = (1 / N2) * (1 / N2) / n;
    se = sqrt(varN2 / n);
    fprintf(salida, ("SE(N2) = " + FloatToStrF(se, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Bandas de confianza al 95 %\n");
    fprintf(salida, (FloatToStrF( 1 / (1 / N2 + TDeStudent(n-1) * se), ffFixed, 3, 3) +
"\t" + FloatToStrF( 1 / (1 / N2 - TDeStudent(n-1) * se), ffFixed, 3, 3) + "\n").c_str());
    double y1 = 1 / N1;
    //fprintf(salida, "Recíproco de la densidad \"y(N) = 1 / N\"\n");
    //fprintf(salida, ("y(N1) = " + FloatToStrF(y1, ffFixed, 3, 3) + "\n").c_str());
    double y2 = 1 / N2;
    //fprintf(salida, "Recíproco de la densidad \"y(N) = 1 / N\"\n");
    //fprintf(salida, ("y(N2) = " + FloatToStrF(y2, ffFixed, 3, 3) + "\n").c_str());
    double vary1 = y1 * y1 / n;
    fprintf(salida, "Varianza del recíproco de la densidad (4.13)\n");
    fprintf(salida, ("Var(y(N1)) = " + FloatToStrF(vary1, ffFixed, 3, 3) + "\n").c_str());
    double vary2 = y2 * y2 / n;
    fprintf(salida, "Varianza del recíproco de la densidad (4.13)\n");
    fprintf(salida, ("Var(y(N2)) = " + FloatToStrF(vary2, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Error estandar\n");
    fprintf(salida, ("SE(y(N1)) = " + FloatToStrF(sqrt(vary1 / n), ffFixed, 3, 3) +
"\n").c_str());
    fprintf(salida, "Error estandar\n");
    fprintf(salida, ("SE(y(N2)) = " + FloatToStrF(sqrt(vary2 / n), ffFixed, 3, 3) + "\n").c_str());
    double N3 = sqrt(N1 * N2);
    fprintf(salida, "Estimado de la densidad (4.14)\n");
    fprintf(salida, "(Estimador de Diggle)\n");
    fprintf(salida, ("N3 = " + FloatToStrF(N3, ffFixed, 3, 3) + "\n").c_str());
    double varN3 = (1 / N3) * (1 / N3) / n;
    fprintf(salida, "Varianza de la densidad (4.15)\n");
    fprintf(salida, ("Var(N3) = " + FloatToStrF(varN3, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Error estandar\n");
    se = sqrt(varN3 / n);
    fprintf(salida, ("SE(N3) = " + FloatToStrF(se, ffFixed, 3, 3) + "\n").c_str());
    fprintf(salida, "Bandas de confianza al 95 %\n");
    fprintf(salida, (FloatToStrF( 1 / (1 / N3 + TDeStudent(n-1) * se), ffFixed, 3, 3) +
"\t" + FloatToStrF( 1 / (1 / N3 - TDeStudent(n-1) * se), ffFixed, 3, 3) + "\n").c_str());

    fclose(salida);

Graphics::TBitmap *grafica = new Graphics::TBitmap();
grafica->Width = maxX / 10;
grafica->Height = maxY / 10;

if(CheckBoxDelimitarArea->Checked)
    this->dibujarCirculos(cantidadCirculos, circulosX, circulosY, grafica, radio);
this->dibujarMuestras(cantidad, coordX, coordY, grafica);

```

```

        grafica->Canvas->Pen->Color = clBlack;
        grafica->Canvas->Pen->Width = 2;
        for(int y = 0; y < n; y++){
            grafica->Canvas->PenPos = Point((cuadradoX[y] - tamanoCuadrado) / 10, grafica-
            >Height - (cuadradoY[y] - tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + tamanoCuadrado) / 10, grafica->Height -
            (cuadradoY[y] - tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] + tamanoCuadrado) / 10, grafica->Height -
            (cuadradoY[y] + tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - tamanoCuadrado) / 10, grafica->Height -
            (cuadradoY[y] + tamanoCuadrado) / 10);
            grafica->Canvas->LineTo((cuadradoX[y] - tamanoCuadrado) / 10, grafica->Height -
            (cuadradoY[y] - tamanoCuadrado) / 10);
        }
        grafica->SaveToFile("mapa muestreo Byth y Ripley.bmp");
        delete grafica;
    }

    ShowMessage("Listo.");
}
//-----

void __fastcall TForm1::ButtonDibujarClick(TObject *Sender)
{
    Graphics::TBitmap *salida = new Graphics::TBitmap();
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r");
    char *buffer = (char*)malloc(sizeof(char) * 100);
    AnsiString linea;
    int x, y;

    salida->Width = 1000;
    salida->Height = 1000;
    salida->Canvas->Pen->Color = clRed;
    salida->Canvas->Pen->Width = 3;

    while(! feof(entrada)){
        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
        if (linea.Length() > 0){
            x = (RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1))) / 10;
            y = (RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000))) / 10;
            y = 1000 - y;
            salida->Canvas->PenPos = Point(x, y);
            salida->Canvas->LineTo(x + 2, y + 2);
        }
    }
    salida->SaveToFile("mapa muestreo.bmp");

    fclose(entrada);
    free(buffer);
}
//-----

void __fastcall TForm1::ButtonTSquareClick(TObject *Sender)
{
    //leer los datos del archivo
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r"),
        *salida, *circulos;
    //log = fopen("log.txt", "w+");
    char *buffer = (char*)malloc(sizeof(char) * 100);
    AnsiString linea;

    double coordX[1000], coordY[1000], swap, distancia,
        maxX = 0, maxY = 0, xx[100], zz[100];
    int cantidad = 0, cantidadCirculos = 0, indice1, indice2, n, cercano, cercano2, nreal;
    double hipotenusa, angulo2, radio;
    bool ok, elegidos[1000];

    for(int x = 0; x < 1000; x++)
        elegidos[x] = false;

```

```

if(CheckBoxArchivoDeCirculos->Checked){
    //los centros de los circulos son leidos de un archivo
    //guardo en coordX y coordY esos centros, guardo en cantidadCirculos la cantidad
    //de circulos leidos
    circulos = fopen(EditArchivoDeCirculos->Text.c_str(), "r");
    while(! feof(circulos)){
        RU_LeerLinea(circulos, buffer);
        linea = AnsiString(buffer);
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
            cantidad++;
        }
    }
    cantidadCirculos = cantidad;
    fclose(circulos);
}

while(! feof(entrada)){
    //leo las muestras
    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    if(linea.Length() > 0){
        coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
        if(coordX[cantidad] > maxX)
            maxX = coordX[cantidad];
        coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
        if(coordY[cantidad] > maxY)
            maxY = coordY[cantidad];
        cantidad++;
    }
}

if(! CheckBoxArchivoDeCirculos->Checked)
    //si los centros no se leyeron de archivo, entonces se toman como
    //centros de circulos las muestras leidas
    cantidadCirculos = 0;

maxX+= 100; //limites del area de trabajo
maxY+= 100; //limites del area de trabajo

fclose(entrada);
free(buffer);

//ordenar aleatoriamente los valores para elegir los primeros n valores
//para evitar que un arbol sea elegido mas de una vez
randomize();
/* for(int x = 1; x <= 500; x++){
    indice1 = random(cantidad - cantidadCirculos) + cantidadCirculos;
    indice2 = random(cantidad - cantidadCirculos) + cantidadCirculos;

    swap = coordX[indice1];
    coordX[indice1] = coordX[indice2];
    coordX[indice2] = swap;

    swap = coordY[indice1];
    coordY[indice1] = coordY[indice2];
    coordY[indice2] = swap;
}*/

Graphics::TBitmap *grafica = new Graphics::TBitmap();
grafica->Width = maxX / 10;
grafica->Height = maxY / 10;

n = StrToInt(EditN->Text);
if(CheckBoxDelimitarArea->Checked){
    if(RadioButtonOptimo->Checked)
        radio = this-
>delimitarAreaConCirculos((cantidadCirculos>0)?cantidadCirculos:cantidad, coordX, coordY,
grafica);
    else{

```

```

        radio = StrToFloat(EditRadio->Text);
    }
}

if(CheckBoxDelimitarArea->Checked)
    this->dibujarCirculos(cantidadCirculos, coordX, coordY, grafica, radio);
this->dibujarMuestras(cantidad - cantidadCirculos, coordX + cantidadCirculos, coordY +
cantidadCirculos, grafica);

grafica->Canvas->Pen->Width = 2;

//Metodo de T-Square. Sección 4.2.2 de Ecological Methodology. CJ Krebs
//busco n puntos al azar, busco el arbol mas cercano y su vecino mas cercano
int pasada = 0;
nreal = n;
for(int x = 0; x < n; x++){
    Label8->Caption = IntToStr(x) + "      ";
    Label8->Repaint();
    pasada++;
    Label9->Caption = IntToStr(pasada) + "      ";
    Label9->Repaint();
    indice1 = random(maxX - 1);
    indice2 = random(maxY - 1);

    xx[x] = 100000;
    cercano = -1;
    ok = false;

    if(CheckBoxDelimitarArea->Checked)
        //veo si el punto elegido al azar cayo dentro de un circulo
        for (int y = 0; y < ((cantidadCirculos>0)?cantidadCirculos:cantidad); y++){
            distancia = sqrt(powl((indice1 - coordX[y]), 2) + powl(indice2 - coordY[y], 2));
            if(distancia < radio)
                ok = true;
        }

    //busco el arbol mas cercano al punto elegido al azar
    for (int y = cantidadCirculos; (y < cantidad) && ok; y++){
        distancia = sqrt(powl((indice1 - coordX[y]), 2) + powl(indice2 - coordY[y], 2));

        ok = true;
        if(CheckBoxDelimitarArea->Checked){
            if(distancia > radio)
                ok = false; //no esta dentro del circulo del punto y
            else
                ok = this->estaLaRectaDentroDelArea(indice1, indice2, coordX[y], coordY[y],
coordX, coordY, (cantidadCirculos>0)?cantidadCirculos:cantidad, radio);
        }
        if( (distancia < xx[x]) && ok){
            xx[x] = distancia;
            cercano = y;
        }
        ok = true;
    }

    //si el mas cercano ya fue elegido no lo analizo nuevamente
    if(cercano >= 0){
        if(elegidos[cercano])
            cercano = -1;
        else
            elegidos[cercano] = true;
    }

    zz[x] = 100000;
    //busco el vecino mas cercano tal que cumpla la condicion T-Square
    for (int y = cantidadCirculos; (y < cantidad) && (cercano >= 0); y++){
        if (y != cercano){
            distancia = sqrt(powl((coordX[cercano] - coordX[y]), 2) + powl(coordY[cercano] -
coordY[y], 2));
            if(distancia < zz[x]){
                //controlo que el arbol y cumpla la condicion T-Square

```

```

        try{
            hipotenusa = sqrt(pow1((indice1 - coordX[y]), 2) + pow1(indice2 - coordY[y],
2));
            angulo2 = (hipotenusa * hipotenusa - distancia * distancia - xx[x] * xx[x]) /
((-2) * distancia * xx[x]);
            angulo2 = acos(angulo2);
            angulo2 = angulo2 * 90 / (pi/2);
        }
        catch(Exception &e){
            ShowMessage("Error calculando el ángulo");
        }

        ok = true;
        if(CheckBoxDelimitarArea->Checked)
            ok = this->estaLaRectaDentroDelArea(coordX[cercano], coordY[cercano],
coordX[y], coordY[y], coordX, coordY, (cantidadCirculos>0)?cantidadCirculos:cantidad,
radio);

        if((angulo2 >= 90) && ok){
            zz[x] = distancia;
            cercano2 = y;
            pasada = 0;
        }
    }
}

if( (zz[x] > 90000) || (xx[x] > 90000) )
    x--; //no encuentre un arbol crecano que cumpla la condicion T-Square
        //hago otra seleccion al azar y busco nuevamente
    else{
        //fprintf(log, (FloatToStr(indice1) + " ; " + FloatToStr(indice2) +
"\n").c_str());
        //fprintf(log, (FloatToStr(coordX[cercano]) + " ; " + FloatToStr(coordY[cercano])
+ "\n").c_str());
        //fprintf(log, (FloatToStr(coordX[cercano2]) + " ; " +
FloatToStr(coordY[cercano2]) + "\n").c_str());
        grafica->Canvas->PenPos = Point((indice1) / 10, grafica->Height - (indice2) / 10);
        grafica->Canvas->Pen->Color = clPurple;
        grafica->Canvas->LineTo((coordX[cercano]) / 10, grafica->Height -
(coordY[cercano]) / 10);
        grafica->Canvas->Pen->Color = clBlue;
        grafica->Canvas->LineTo((coordX[cercano2]) / 10, grafica->Height -
(coordY[cercano2]) / 10);
    }

    if(pasada > 500){
        ok = MessageDlg("Seguir con otros 500?", mtConfirmation, botonesSiNo, 0) == mrYes;
        if(ok)
            pasada = 0;
        else{
            nreal = x - 1;
            x = n;
        }
    }
}

if(SaveDialog2->Execute())
    grafica->SaveToFile(SaveDialog2->FileName);

//----
/*n = 17;
FILE *temporal = fopen("temp.txt", "r");
for(int x = 0; x < n; x++){
    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    xx[x] = StrToFloat(linea);

    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    zz[x] = StrToFloat(linea);
}

```

```

    }
    fclose(temporal);*/

    if(SaveDialog1->Execute()){
        //escribo la salida
        n = nreal;
        salida = fopen(SaveDialog1->FileName.c_str(), "w+");
        fprintf(salida, ("N = " + IntToStr(n) + "\n").c_str());
        if(CheckBoxDelimitarArea->Checked)
            fprintf(salida, ("Radio de círculos = " + FloatToStrF(radio, ffFixed, 3, 3) +
"\n").c_str());

        double factor = StrToFloat(EditFactor->Text);

        double acum1 = 0, acums1 = 0;
        double acum2 = 0, acums2 = 0;
        double sx = 0, sz = 0, sxz = 0;
        fprintf(salida, "i\txi\tzi\n");
        for(int x = 0; x < n; x++){
            fprintf(salida, (IntToStr(x+1) + "\t" + FloatToStrF(xx[x], ffFixed, 3, 3) + "\t" +
FloatToStrF(zz[x], ffFixed, 3, 3) + "\n").c_str());

            acum1+= xx[x] * xx[x];
            acums1+= xx[x];
            acum2+= zz[x] * zz[x];
            acums2+= zz[x];
            sxz+= xx[x] * zz[x];
        }

        fprintf(salida, "Test de Hines and Hines (4.16)\n");
        double ht = 2*n*(2 * acum1 + acum2) / powl(sqrt(2) * acums1 + acums2, 2);
        fprintf(salida, ("ht = " + FloatToStrF(ht, ffFixed, 3, 3) + "\n").c_str());
        struct ParDeValores critical = criticalValuesForHinesTest(n, ALFA_05);
        if(ht < critical.value1)
            fprintf(salida, ("El ht es menor que " + FloatToStrF(critical.value1, ffFixed, 3, 3)
+ ", por lo tanto los datos estan distribuidos de manera uniforme.\n").c_str());
        else
            if(ht > critical.value2)
                fprintf(salida, ("El ht es mayor que " + FloatToStrF(critical.value2, ffFixed,
3, 3) + ", por lo tanto los datos estan agrupados.\n").c_str());
            else
                fprintf(salida, ("El ht es mayor que " + FloatToStrF(critical.value1, ffFixed,
3, 3) + " y menor que " + FloatToStrF(critical.value2, ffFixed, 3, 3) + ", por lo tanto
los datos tienen una distribución al azar.\n").c_str());

        double N4 = 2*n / (pi * acum2) * factor;
        fprintf(salida, "Estimado de la densidad (4.17)\n");
        fprintf(salida, ("N4 = " + FloatToStrF(N4, ffFixed, 3, 3) + "\n").c_str());
        double Nt = n*n / (2 * acums1 * sqrt(2) * acums2) * factor;
        fprintf(salida, "Densidad media estimada (Nt) (4.18)\n");
        fprintf(salida, ("Nt = " + FloatToStrF(Nt, ffFixed, 3, 3) + "\n").c_str());

        sx = (acum1 - acums1 * acums1 / n) / (n-1);
        sz = (acum2 - acums2 * acums2 / n) / (n-1);
        sxz = (sxz - acums1 * acums2 / n) / (n-1);
        fprintf(salida, "Error estandar (4.19)\n");
        double se = sqrt ( 8 * ((acums2/n) * (acums2/n) * sx + 2* (acums1/n) * (acums2/n) *
sxz + (acums1/n) * (acums1/n) * sz) / n) / factor;
        fprintf(salida, ("SE(Nt) = " + FloatToStrF(se, ffFixed, 3, 3) + "\n").c_str());
        fprintf(salida, "Bandas de confianza al 95 %\n");
        fprintf(salida, (FloatToStrF(1 / (1 / Nt + TDeStudent(n-1) * se), ffFixed, 3, 3) +
"\t" + FloatToStrF(1 / (1 / Nt - TDeStudent(n-1) * se), ffFixed, 3, 3) + "\n").c_str());

        fclose(salida);
    }

    //fclose(log);
    ShowMessage("Listo");
}
//-----

```



```

void __fastcall TForm1::ButtonExaminarClick(TObject *Sender)
{
    if(OpenDialog1->Execute()){
        EditArchivo->Text = OpenDialog1->FileName;
    }
}
//-----

void __fastcall TForm1::EditArchivoChange(TObject *Sender)
{
    archivoTrabajo = EditArchivo->Text;
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    //leer los datos del archivo
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r");
    char *buffer = (char*)malloc(sizeof(char) * 100);
    AnsiString linea;

    double coordX[100], coordY[100], maxX = 0, maxY = 0;
    int cantidad = 0;
    double radio;

    while(! feof(entrada)){
        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            if(coordX[cantidad] > maxX)
                maxX = coordX[cantidad];
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 100));
            if(coordY[cantidad] > maxY)
                maxY = coordY[cantidad];
            cantidad++;
        }
    }

    maxX+= 100; //limites del area de trabajo
    maxY+= 100; //limites del area de trabajo

    fclose(entrada);
    free(buffer);
    Graphics::TBitmap *grafica = new Graphics::TBitmap();
    radio = this->delimitarAreaConCirculos(cantidad, coordX, coordY, grafica);

    if(SaveDialog1->Execute()){
        this->dibujarCirculos(cantidad, coordX, coordY, grafica, radio);
        grafica->SaveToFile(SaveDialog1->FileName);
    }

    ShowMessage("Listo.");
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    if(RadioButtonAlazar->Checked)
        this->cuadratasAlazar();
    else
        this->cuadratasPorGrilla();
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    botonesSiNo << mbYes;
    botonesSiNo << mbNo;
}
//-----

```

```

bool estaEnElArea(Punto *punto, double *circulosX, double *circulosY, int
cantidadDeCirculos, double radio){
    bool ok = false;
    double distancia = radio * 3;
    for (int y = 0; (y < cantidadDeCirculos) && ! ok; y++){
        distancia = sqrt(powl((punto->x - circulosX[y]), 2) + powl(punto->y -
circulosY[y], 2));
        if(distancia < radio)
            ok = true;
    }
    return ok;
}

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    AnsiString linea, figura = "cuadrado";
    long double lado1, angulo, angulo2;
    int cuadratas, cantidadCaquitas, mosaico2, cantidadMosaico, cantidadMosaico2, index,
cantidad;
    Punto *punto1 = new Punto(), *punto2 = new Punto(),
        *punto3 = new Punto(), *punto4 = new Punto(), *punto5 = new Punto();
    Graphics::TBitmap *laguna = new Graphics::TBitmap();
    bool ok, flag, cancelar = false;
    TablaDeFrecuencia *frecuencia = new TablaDeFrecuencia();
    OrderedCollection *cuadrados = new OrderedCollection();
    Figura *cuad = new Cuadrado(), *cuad2;
    long double xmin, xmax, ymin, ymax, coordX[100], coordY[100];
    FILE *entrada = fopen(archivoTrabajo.c_str(), "r"),
        *salida;
    char *buffer = (char*)malloc(sizeof(char) * 100);
    PuntoMapeado *mosaico, *indiceMosaico, *indiceDeX;
    double offsetX = 160;
    double offsetY = 150;

    cantidad = 0;
    RU_LeerLinea(entrada, buffer);
    linea = AnsiString(buffer);
    xmax = xmin = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1)) + offsetX;
    ymax = ymin = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000)) + offsetY;
    while(! feof(entrada)){
        if(linea.Length() > 0){
            coordX[cantidad] = RU_StrToFloat(linea.SubString(1, linea.AnsiPos(";") - 1));
            coordX[cantidad] += offsetX;
            if(coordX[cantidad] > xmax)
                xmax = coordX[cantidad];
            if(coordX[cantidad] < xmin)
                xmin = coordX[cantidad];
            coordY[cantidad] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") + 1, 1000));
            coordY[cantidad] += offsetY;
            if(coordY[cantidad] > ymax)
                ymax = coordY[cantidad];
            if(coordY[cantidad] < ymin)
                ymin = coordY[cantidad];
            cantidad++;
        }

        RU_LeerLinea(entrada, buffer);
        linea = AnsiString(buffer);
    }
    fclose(entrada);

    xmax += offsetX;
    ymax += offsetY;

    //armo el mosaico
    int paso = 1;
    while((xmax - xmin) / paso > 500)
        paso++;
    while((ymax - ymin) / paso > 500)
        paso++;
}

```

```

int longitudMosaico = ((xmax - xmin + 1) / paso) * ((ymax - ymin + 1) / paso);
while(longitudMosaico < 0){
    paso++;
    longitudMosaico = ((xmax - xmin + 1) / paso) * ((ymax - ymin + 1) / paso);
}
mosaico = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (longitudMosaico + 1));
indiceMosaico = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (longitudMosaico + 1));
indiceDeX = (PuntoMapeado*)malloc(sizeof(PuntoMapeado) * (((xmax - xmin + 1) / 2) + 1));
//indiceDeX guarda las posiciones del vector mosaico donde comienzan las
columnas de la matriz
laguna->Height = ymax / paso * 2;
laguna->Width = xmax / paso * 2 + 55;

int cantidadDeCirculos = 0;
double circulosX[1000], circulosY[1000];
//los centros de los circulos son leidos de un archivo
//guardo en coordX y coordY esos centros, guardo en cantidaCirculos la cantidad
//de circulos leidos
FILE *circulos = fopen(EditArchivoDeCirculos->Text.c_str(), "r");
while(! feof(circulos)){
    RU_LeerLinea(circulos, buffer);
    linea = AnsiString(buffer);
    if(linea.Length() > 0){
        circulosX[cantidadDeCirculos] = RU_StrToFloat(linea.SubString(1,
linea.AnsiPos(";") - 1));
        circulosY[cantidadDeCirculos] = RU_StrToFloat(linea.SubString(linea.AnsiPos(";") +
1, 1000));
        cantidadDeCirculos++;
    }
}
fclose(circulos);

double radio = StrToInt(EditRadio->Text);

//dibujo el perímetro
this->dibujarCirculos(cantidadDeCirculos, circulosX, circulosY, laguna, radio);
//dibujo los individuos
for(int x = 0; x < cantidad; x++){
    laguna->Canvas->Pen->Color = clRed;
    laguna->Canvas->Pen->Width = 3;
    laguna->Canvas->PenPos = Point(coordX[x] / paso * 2, laguna->Height - coordY[x] / paso
* 2);
    laguna->Canvas->LineTo(coordX[x] / paso * 2 + 1, laguna->Height - coordY[x] / paso * 2
+ 1);
    //laguna->Canvas->TextOut(coordX[x] / paso * 2, laguna->Height - coordY[x] / paso * 2,
IntToStr(x));
}

//inicializo el mapeo de puntos
cantidadMosaico2 = 0;
indicex = 0;
for(int x = xmin; x <= xmax; x+= paso){
    Label8->Caption = IntToStr(x) + "          ";
    Label8->Repaint();
    indiceDeX[indicex].x = x;
    indiceDeX[indicex].indice = cantidadMosaico2;
    indicex++;
    for(int y = ymin; y <= ymax; y+= paso){
        puntol->x = x;
        puntol->y = y;
        if(estaEnElArea(puntol, circulosX, circulosY, cantidadDeCirculos, radio)){
            mosaico[cantidadMosaico2].x = x;
            mosaico[cantidadMosaico2].y = y;
            mosaico[cantidadMosaico2].indice = cantidadMosaico2;
            indiceMosaico[cantidadMosaico2].indice = cantidadMosaico2;
            cantidadMosaico2++;
        }
    }
}
cantidadMosaico = cantidadMosaico2;
indiceDeX[indicex].x = -1;

```

```

//inicializo variables
frecuencia->vaciar();
cuadrados->clear();
flag = true;

//inicializo el lado del cuadrado y la cantidad de cuadratas
lado1 = StrToFloat(EditTamanoDeLado->Text);
cuadratas = StrToFloat(EditNCuadratas->Text);

int corte = 0;
int nreal = cuadratas;

randomize();

cantidadMosaico = cantidadMosaico2;
for(int y = 0; y < cantidadMosaico; y++)
    indiceMosaico[y].x = 0;

    ProgressBar1->Position = 0;
    ProgressBar1->Max = cuadratas;
    ProgressBar2->Position = 0;
    ProgressBar2->Max = cantidadMosaico;
    for(int y = 1; (y <= cuadratas) && flag; y++){
        ProgressBar1->StepIt();
        Label8->Caption = "Colocando cuadrata no. " + IntToStr(y) + "      ";
        Label8->Repaint();

        ok = true;
        while(ok){
            if (cancelar)
                return;
            ProgressBar2->StepIt();
            Label8->Repaint();
            Label9->Repaint();

            //armar el cuadrado dentro del area de muestreo
            //elijo el punto1
            //Para esto elijo un mosaico al azar

            mosaico2 = random(cantidadMosaico);
            flag = true;

            punto1->x = mosaico[indiceMosaico[mosaico2].indice].x;
            punto1->y = mosaico[indiceMosaico[mosaico2].indice].y;

            while(flag && ! estaEnElArea(punto1, circulosX, circulosY,
cantidadDeCirculos, radio)){
                //elimino el mosaico 'mosaico2'
                indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
                indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
                indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
                //actualizo el indice en mosaico
                mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
                mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
                cantidadMosaico--;

                mosaico2 = random(cantidadMosaico);
                if(cantidadMosaico == 0)
                    flag = false;
                flag = cantidadMosaico != 0;
                punto1->x = mosaico[indiceMosaico[mosaico2].indice].x;
                punto1->y = mosaico[indiceMosaico[mosaico2].indice].y;
            }

            //elijo el punto2 (a partir de un angulo)
            angulo = random(90); //random(360);

            for(int rotaciones = 1; rotaciones < 90; rotaciones+= 5){

                if(angulo < 90){

```

```

punto2->x = punto1->x + cosl(angulo * pi / 180) * lado1;
punto2->y = punto1->y + sinl(angulo * pi / 180) * lado1;
}
else
    if(angulo < 180){
        angulo2 = 180 - angulo;
        punto2->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
        punto2->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
    }
    else
        if(angulo < 270){
            angulo2 = angulo - 180;
            punto2->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
            punto2->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
        }
        else{
            angulo2 = 360 - angulo;
            punto2->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
            punto2->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
        }
    //si el punto2 no esta en el area de muestreo lo giro 180 grados tomando
    como eje el punto x1
    if(! estaEnElArea(punto2, circulosX, circulosY, cantidadDeCirculos, radio)){
        punto2->x = punto1->x - (punto2->x - punto1->x);
        punto2->y = punto1->y - (punto2->y - punto1->y);
    }
    //calculo los punto3 y punto4

    if(punto1->x == punto2->x){
        punto3->x = punto4->x = punto1->x - lado1;
        punto3->y = punto1->y;
        punto4->y = punto2->y;
        if((! estaEnElArea(punto3, circulosX, circulosY, cantidadDeCirculos,
radio)) || (! estaEnElArea(punto4, circulosX, circulosY, cantidadDeCirculos, radio))){
            //volteo el cuadrado tomando el eje punto1-punto2
            punto3->x = punto4->x = punto1->x + lado1;
        }
    }
    else
        if(punto1->x < punto2->x){
            if(angulo < 180){
                angulo2 = 90 - angulo;
                punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
                if((! estaEnElArea(punto3, circulosX, circulosY, cantidadDeCirculos,
radio)) || (! estaEnElArea(punto4, circulosX, circulosY, cantidadDeCirculos, radio))){
                    //volteo el cuadrado tomando el eje punto1-punto2
                    punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                    punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                    punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
                    punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
                }
            }
            else{
                angulo2 = angulo - 270;
                punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
                if((! estaEnElArea(punto3, circulosX, circulosY,
cantidadDeCirculos, radio)) || (! estaEnElArea(punto4, circulosX, circulosY,
cantidadDeCirculos, radio))){
                    //volteo el cuadrado tomando el eje punto1-punto2
                    punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
                    punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                    punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
                    punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
                }
            }
        }
    }
}

```

```

    }
    else{
        if(angulo < 90){
            angulo2 = 90 - angulo;
            punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
            if((! estaEnElArea(punto3, circulosX, circulosY,
cantidadDeCirculos, radio)) || (! estaEnElArea(punto4, circulosX, circulosY,
cantidadDeCirculos, radio))){
                //volteo el cuadrado tomando el eje punto1-punto2
                punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
            }
        }
        else{
            angulo2 = 270 - angulo;
            punto3->x = punto1->x - cosl(angulo2 * pi / 180) * lado1;
            punto3->y = punto1->y + sinl(angulo2 * pi / 180) * lado1;
            punto4->x = punto2->x - cosl(angulo2 * pi / 180) * lado1;
            punto4->y = punto2->y + sinl(angulo2 * pi / 180) * lado1;
            if((! estaEnElArea(punto3, circulosX, circulosY,
cantidadDeCirculos, radio)) || (! estaEnElArea(punto4, circulosX, circulosY,
cantidadDeCirculos, radio))){
                //volteo el cuadrado tomando el eje punto1-punto2
                punto3->x = punto1->x + cosl(angulo2 * pi / 180) * lado1;
                punto3->y = punto1->y - sinl(angulo2 * pi / 180) * lado1;
                punto4->x = punto2->x + cosl(angulo2 * pi / 180) * lado1;
                punto4->y = punto2->y - sinl(angulo2 * pi / 180) * lado1;
            }
        }
    }

    ok = false;
    ((Cuadrado*)cuad)->setPuntos(punto1, punto2, punto3, punto4);
    for(int ww = 0; ww < cuadrados->count; ww++){
        if( ((Cuadrado*)cuad)->intersectasCon(((Cuadrado*)(cuadrados->items[ww])) )
            ok = true;
        }
    }
    // if(ok){
    //     Cuadrado *nuevo = new Cuadrado(punto1, punto2, punto3, punto4);
    //     cuadrados->add(nuevo);
    // }

    /*int maxx = max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)), maxy
= max(max(punto1->y, punto2->y), max(punto3->y, punto4->y));
    int minx = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)), miny =
min(min(punto1->y, punto2->y), min(punto3->y, punto4->y));
    minx-= xmin; miny-= ymin;
    minx = minx / paso; miny = miny / paso;
    minx = minx * paso + xmin; miny = miny * paso + ymin;
    for(int ww = minx; (ww <= maxx) && ! ok; ww+= paso){
        //busco en el indice de x
        indicex = 0;
        while((indiceDeX[indicex].x != ww) && (indiceDeX[indicex].x != -1))
            indicex++;
        indicex = indiceDeX[indicex].indice;
        for(int zz = miny; (zz <= maxy) && ! ok; zz+= paso){
            punto5->x = ww;
            punto5->y = zz;
            if(cuad->tePerteneceElPunto(punto5)){
                //busco en mosaico el indice de y
                while((mosaico[indicex].y != zz) && (mosaico[indicex].x == ww))
                    indicex++;

                if(mosaico[indicex].x == ww){
                    ok = (indiceMosaico[mosaico[indicex].indice].x == 1);
                }
            }
        }
    }

```

```

    }
  }*/
  angulo = int(angulo + 5) % 90;

  if(! ok)
    rotaciones = 90;
  } //del for rotaciones

  //ok = subarea->intersectaAlgunAreaAnalizada((Cuadrado*)cuad = new
Cuadrado(punto1, punto2, punto3, punto4));
  ok = ok || ! estaEnElArea(punto1, circulosX, circulosY, cantidadDeCirculos,
radio);
  ok = ok || ! estaEnElArea(punto2, circulosX, circulosY, cantidadDeCirculos,
radio);
  ok = ok || ! estaEnElArea(punto3, circulosX, circulosY, cantidadDeCirculos,
radio);
  ok = ok || ! estaEnElArea(punto4, circulosX, circulosY, cantidadDeCirculos,
radio);

  if(ok){
    //si luego de 20 intentos de armar un cuadrado a partir de
indiceMosaico[mosaico2] no lo logré
    //elimino el mosaico.
    indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
    indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
    indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
    //actualizo el indice en mosaico
    mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
    mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
    cantidadMosaico--;

    if(cantidadMosaico == 0)
      flag = false;
  }

  if(cantidadMosaico == 0)
    flag = false;
  flag = cantidadMosaico != 0;
  if(! flag)
    ok=false; //salgo a la fuerza. No hay lugar para poner cuadrados

  corte++;
  Label9->Caption = "Corte " + IntToStr(corte) + " ";
  Label9->Repaint();
  if(corte == 500){
    ok = MessageDlg("Seguir con otros 500?", mtConfirmation, botonesSiNo, 0) ==
mrYes;

    if(ok)
      corte = 0;
    else{
      nreal = y - 1;
      y = cuadratas;
    }
  }
} //del while ok

if(flag){
  //elimino el mosaico 'mosaico2'
  //mosaico[mosaico2].x = -1;
  indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
  indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
  indiceMosaico[cantidadMosaico] = indiceMosaico[longitudMosaico];
  //actualizo el indice en mosaico
  mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
  mosaico[indiceMosaico[cantidadMosaico].indice].indice = cantidadMosaico;
  cantidadMosaico--;
  if(cantidadMosaico == 0)
    flag = false;

  cuadrados->add(cuad2 = new Cuadrado(punto1, punto2, punto3, punto4));

```

```
//eliminar los mosaicos que estan dentro del cuadrado
ok = true;
int maxx = max(max(punto1->x, punto2->x), max(punto3->x, punto4->x)), maxy = 
max(max(punto1->y, punto2->y), max(punto3->y, punto4->y));
int minx = min(min(punto1->x, punto2->x), min(punto3->x, punto4->x)), miny = 
min(min(punto1->y, punto2->y), min(punto3->y, punto4->y));
minx-= xmin; miny-= ymin;
minx = minx / paso; miny = miny / paso;
minx = minx * paso + xmin; miny = miny * paso + ymin;
for(int ww = minx; ww <= maxx; ww+= paso)
    for(int zz = miny; zz <= maxy; zz+= paso){
        punto5->x = ww;
        punto5->y = zz;
        if(cuad2->tePerteneceElPunto(punto5)){
            //busco en el indice de x
            indicex = 0;
            while((indiceDeX[indicex].x != ww) && (indiceDeX[indicex].x != -1))
                indicex++;
            indicex = indiceDeX[indicex].indice;
            //busco en mosaico el indice de y
            while((mosaico[indicex].y != zz) && (mosaico[indicex].x == ww) &&
(indicex < longitudMosaico))
                indicex++;
            if(mosaico[indicex].x == ww){
                mosaico2 = mosaico[indicex].indice;
                indiceMosaico[longitudMosaico] = indiceMosaico[mosaico2];
                indiceMosaico[mosaico2] = indiceMosaico[cantidadMosaico];
                indiceMosaico[cantidadMosaico] = indiceMosaico[indicex];
                //marco el mosaico como "usado" por un cuadrado
                indiceMosaico[cantidadMosaico].x = 1;
                //actualizo el indice en mosaico
                mosaico[indiceMosaico[mosaico2].indice].indice = mosaico2;
                mosaico[indiceMosaico[cantidadMosaico].indice].indice = 
cantidadMosaico;
                cantidadMosaico--;
                if(cantidadMosaico == 0)
                    flag = false;
                ProgressBar2->StepIt();
            }
        }
    }

//ver caquitas dentro de area
cantidadCaquitas = 0;
for(int x = 0; x < cantidad; x++){
    punto5->x = coordX[x];
    punto5->y = coordY[x];
    if(cuad2->tePerteneceElPunto(punto5))
        cantidadCaquitas++;
}
if(corte < 500)
    frecuencia->agregar(cantidadCaquitas);

//*****
//Dibujo el perimetro en un bitmap
if((figura == "cuadrado") && (corte < 500)){
    laguna->Canvas->Pen->Color = clBlack;
    laguna->Canvas->Pen->Width = 2;
    laguna->Canvas->PenPos = Point((punto1->x) / paso * 2, laguna->Height -
(punto1->y) / paso * 2);
    laguna->Canvas->LineTo((punto2->x) / paso * 2, laguna->Height - (punto2->y) /
paso * 2);
    laguna->Canvas->LineTo((punto4->x) / paso * 2, laguna->Height - (punto4->y) /
paso * 2);
    laguna->Canvas->LineTo((punto3->x) / paso * 2, laguna->Height - (punto3->y) /
paso * 2);
    laguna->Canvas->LineTo((punto1->x) / paso * 2, laguna->Height - (punto1->y) /
paso * 2);
}

}

}
```



```

        } //del for de 1 hasta cuadratas
    //} //del for de 1 hasta mosaicos

    laguna->SaveToFile("Mapa cuadratas al azar.bmp");
    laguna->FreeImage();
    delete laguna;

    if(SaveDialog1->Execute()){
        //escribo la salida
        salida = fopen(SaveDialog1->FileName.c_str(), "w+");
        fprintf(salida, ("N = " + IntToStr(nreal) + "\n").c_str());
        fprintf(salida, ("Tamano de cuadrado: " + FloatToStr(ladol) + " x " +
FloatToStr(ladol) + "\n").c_str());
        fprintf(salida, "\nTabla de frecuencia\n");
        for(int x = 0; x <= frecuencia->cantidadMaxima(); x++)
            fprintf(salida, (IntToStr(x) + "\t" + IntToStr(frecuencia->frecuenciaDe(x)) +
"\n").c_str());
        fclose(salida);
    }
    free(mosaico);
    free(indiceMosaico);
    free(indiceDeX);
    delete frecuencia;
}
//-----

```

Procedure Uloros.h

```

//-----
#ifndef ULOROSH
#define ULOROSH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Colecciones.hpp>
#include <Dialogs.hpp>
#include <ComCtrls.hpp>
//-----
typedef struct{
    int x, y, indice;
} PuntoMapeado;

class Punto : public TObject{
public:
    long double x, y;

    Punto();
    Punto(long double _x, long double _y);
    Punto(Punto *p);

    double distanciaA(Punto *p);

    Punto* operator - (Punto *p){return new Punto(this->x - p->x, this->y - p->y);}
};

class TablaDeFrecuencia : public TObject{
private:
    int *valores;
    int limite;

public:
    TablaDeFrecuencia();
    void agregar(int cantidad);
    void vaciar();

    int cantidadMaxima();
    int frecuenciaDe(int cantidad);
}

```

```

};

class Figura : public TObject{
public:
    virtual bool intersectasCon(Figura *c) = 0;
    virtual bool tePerteneceElPunto(Punto *p) = 0;
};

class Cuadrado : public Figura{
protected:
    Punto *punto1, *punto2, *punto3, *punto4, *miCentro;

private:
    long double rotacion; //el coseno del angulo
    long double xminrotado, xmaxrotado, yminrotado, ymaxrotado;

public:
    Cuadrado();
    ~Cuadrado();
    Cuadrado(Punto *_punto1, Punto *_punto2, Punto *_punto3, Punto *_punto4);
    void armarEn(TPoint punto, long double _lado, long double _rotacion);
    void setPuntos(Punto *_punto1, Punto *_punto2, Punto *_punto3, Punto *_punto4);

    virtual bool intersectasCon(Figura *c);
    bool existeInterseccion(Cuadrado *c);
    bool tePerteneceElPunto(Punto *p);
    Punto *centro();
    Punto *getPunto1();
    Punto *getPunto2();
    Punto *getPunto3();
    Punto *getPunto4();
    long double productoVectorial(Punto *p1, Punto *p2);
    long double modulo(Punto *p1);
};

class Perinola : public TObject
{
private:
    TRect rectangulo;
    OrderedCollection *perinolas;

public:
    void init(int x1, int y1, int x2, int y2);
    void getPunto(double *x, double *y);
    void eliminarArea(int x1, int y1, int x2, int y2);
};

class TForm1 : public TForm
{
__published:      // IDE-managed Components
    TButton *ButtonBythYRipley;
    TEdit *EditN;
    TButton *ButtonDibujar;
    TLabel *Label1;
    TLabel *Label2;
    TEdit *EditLimiteCuadrado;
    TLabel *Label3;
    TEdit *EditPromedio;
    TButton *ButtonTSquare;
    TEdit *EditFactor;
    TLabel *Label4;
    TEdit *EditArchivo;
    TLabel *Label5;
    TButton *ButtonExaminar;
    TOpenDialog *OpenDialog1;
    TButton *Button1;
    TSaveDialog *SaveDialog1;
    TCheckBox *CheckBoxDelimitarArea;
    TSaveDialog *SaveDialog2;
    TButton *Button2;

```

```

TLabel *Label6;
TLabel *Label7;
TEdit *EditNCuadratas;
TEdit *EditTamanioDeLado;
TRadioButton *RadioButton1;
TRadioButton *RadioButtonAlAzar;
TProgressBar *ProgressBar1;
TProgressBar *ProgressBar2;
TRadioButton *RadioButtonOptimo;
TRadioButton *RadioButton3;
TEdit *EditRadio;
TCheckBox *CheckBoxArchivoDeCirculos;
TEdit *EditArchivoDeCirculos;
TLabel *Label8;
TLabel *Label9;
TLabel *Label10;
TLabel *Label11;
TLabel *Label12;
TLabel *Label13;
TButton *Button3;
void __fastcall ButtonBythYRipleyClick(TObject *Sender);
void __fastcall ButtonDibujarClick(TObject *Sender);
void __fastcall ButtonTSquareClick(TObject *Sender);
void __fastcall ButtonExaminarClick(TObject *Sender);
void __fastcall EditArchivoChange(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);

private:
    AnsiString archivoTrabajo;

    double delimitarAreaConCirculos(int n, double *coordX, double *coordY, Graphics::TBitmap *grafica);
    void dibujarCirculos(int n, double *coordX, double *coordY, Graphics::TBitmap *grafica, double radio);
    void dibujarMuestras(int n, double *coordX, double *coordY, Graphics::TBitmap *grafica);
    void cuadratasAlAzar();
    void cuadratasPorGrilla();

    bool estaEnElAreaDeMuestreo(Punto *p, double x, double y);
    bool estaLaRectaDentroDelArea(double x1, double y1, double x2, double y2, double *coordX, double *coordY, int cantidad,
double radio);

public:
    __fastcall TForm1(TComponent* Owner);

};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```